

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Ingeniería Mecánica



INGENIERÍA INDUSTRIAL

MÁQUINAS Y ESTRUCTURAS INDUSTRIALES

PROYECTO FIN DE CARRERA

***ESTIMACIÓN DEL
ÁNGULO DE DERIVA DE UN VEHÍCULO
MEDIANTE REDES NEURONALES***

AUTOR: D. GUILLERMO MAUDES COMENDADOR

**TUTORAS: Dra. Dña. BEATRIZ LÓPEZ BOADA
Dra. Dña. MARÍA JESÚS LÓPEZ BOADA**

LEGANÉS, 27 JUNIO 2013

Agradecimientos

Con este Proyecto Fin de Carrera termino mi etapa universitaria y quiero agradecer a todos los profesores de la Universidad Carlos III de Madrid que me han impartido clase y han compartido sus conocimientos conmigo para que llegase a ser el Ingeniero que siempre quise ser.

En especial quiero agradecer al departamento de Ingeniería Mecánica al que he estado más unido durante todos estos años y más en particular a la Dra. Dña. Beatriz López Boada y a la Dra. Dña. María Jesús López Boada por haber confiado en mí para llevar a cabo este proyecto y estar siempre disponibles cuando lo necesitaba.

Por último, quiero dar las gracias a todas las personas que han estado a mi lado antes, ahora y siempre, ellos saben quiénes son.



ÍNDICE

1	INTRODUCCIÓN.....	1
1.1	INTRODUCCIÓN.....	3
1.2	OBJETIVOS DEL PROYECTO.....	5
1.3	ESTRUCTURA DEL PROYECTO.....	7
2	SISTEMAS DE SEGURIDAD EN LOS VEHÍCULOS	9
2.1	INTRODUCCIÓN.....	11
2.2	SISTEMAS DE SEGURIDAD.....	13
2.2.1	SEGURIDAD PASIVA.....	13
2.2.2	SEGURIDAD ACTIVA	15
3	DINÁMICA LATERAL	19
3.1	INTRODUCCIÓN.....	21
3.2	CIRCULACIÓN EN CURVA.....	21
3.2.1	VELOCIDAD LIMITE DE DERRAPE	22
3.2.2	VELOCIDAD LIMITE DE VUELCO	23
4	SISTEMA ESP.....	25
4.1	INTRODUCCIÓN.....	27
4.2	FUNCIONAMIENTO	28
4.3	VISIÓN DE FUTURO	32
5	ÁNGULO DE DERIVA DE UN VEHÍCULO	35
5.1	INTRODUCCIÓN.....	37
5.2	DEFINICIÓN	38
5.3	PARÁMETROS DE LOS QUE DEPENDE	42
6	TÉCNICAS PARA LA MEDICIÓN Y ESTIMACIÓN DEL ÁNGULO DE DERIVA.....	45
6.1	INTRODUCCIÓN.....	47
6.2	INTEGRACIÓN DE SENSORES INERCIALES	47
6.3	RELACIONES CINEMÁTICAS	48
6.4	MÉTODOS COMBINADOS.....	52
6.5	SENSORES GPS.....	53
6.6	REDES NEURONALES.....	55
7	REDES NEURONALES ARTIFICIALES	57
7.1	INTRODUCCIÓN.....	59
7.2	DEFINICIÓN	60
7.3	ELEMENTOS BÁSICOS.....	60
7.4	FUNCIONES.....	61
7.4.1	FUNCIÓN DE ENTRADA	61
7.4.2	FUNCIÓN DE ACTIVACIÓN.....	61
7.4.3	FUNCIÓN DE SALIDA.....	63



7.5	MODO DE OPERACIÓN.....	63
7.6	OPERACIONES DE CAPA	64
7.7	OTRAS CONSIDERACIONES	64
7.8	CÁLCULO DEL ERROR.....	64
7.9	MECANISMOS DE APRENDIZAJE	65
7.9.1	APRENDIZAJE SUPERVISADO	66
7.9.2	APRENDIZAJE NO SUPERVISADO.....	67
7.9.3	APRENDIZAJE ON-LINE Y OFF-LINE	68
8	APRENDIZAJE BACKPROPAGATION	69
8.1	INTRODUCCIÓN.....	71
8.2	REGLA DE APRENDIZAJE.....	71
9	SOFTWARE EMPLEADO	79
9.1	INTRODUCCIÓN.....	81
9.2	CARSIM.....	81
9.3	JAVA-NNS	83
10	PROPUESTA DE MODELO DE ESTIMACIÓN DEL ÁNGULO DE DERIVA.....	85
10.1	INTRODUCCIÓN.....	87
10.2	OBTENER LOS PATRONES	87
10.2.1	DEFINICIÓN DE LAS MANIOBRAS.....	90
10.2.1.1	Patrones de entrenamiento	90
10.2.1.2	Patrones de validación.....	94
10.2.2	CONFIGURACIÓN DE LAS MANIOBRAS EN CARSIM	95
10.2.2.1	Configuración del vehículo	95
10.2.2.2	Configuración de la maniobra	95
10.2.2.3	Configuración de los parámetros a extraer.....	97
10.3	TRATAR LOS DATOS	99
10.4	CREAR Y CONFIGURAR LAS REDES.....	101
10.5	ANÁLISIS DE CADA RED	106
10.5.1	Red 1: (4)-(5)-(1)	106
10.5.2	Red 2: (4)-(10)-(1)	108
10.5.3	Red 3: (4)-(15)-(1)	109
10.5.4	Red 4: (4)-(20)-(1)	111
10.5.5	Red 5: (4)-(25)-(1)	112
10.5.6	Red 6: (4)-(5-5)-(1).....	114
10.5.7	Red 7: (4)-(10-10)-(1).....	116
10.6	COMPARATIVA DE LAS REDES.....	117
11	PRESUPUESTO.....	121
11.1	INTRODUCCIÓN.....	123
11.2	PRESUPUESTO DETALLADO	123
11.2.1	PERSONAL.....	123
11.2.2	LICENCIAS Y EQUIPOS DE INFORMÁTICOS.....	123
11.3	PRESUPUESTO GLOBAL.....	124



12 CONCLUSIONES Y TRABAJOS FUTUROS	125
12.1 CONCLUSIONES.....	127
12.2 TRABAJOS FUTUROS.....	129
13 BIBLIOGRAFÍA.....	131
14 ANEXO I.....	139
15 ANEXO II	143
15.1 CARSIM.....	145
15.1.1 MENU 1: CONFIGURACIÓN DEL VEHÍCULO.....	145
15.1.2 MENÚ 2: CONFIGURACIÓN MANIOBRA Y ENTORNO	147
15.1.3 MENÚ 3: REALIZAR LA SIMULACIÓN.....	149
15.1.4 MENÚ 4: VISUALIZACIÓN DE LA SIMULACIÓN Y DE LOS PARÁMETROS.....	149
15.1.5 MENÚ 5: SUPERPONER SIMULACIONES.....	150
15.2 JAVA-NNS	151
15.2.1 CREAR LA CONFIGURACIÓN DE LA RED	151
15.2.2 CREAR LAS CONEXIONES	153
15.2.3 CARGAR PATRONES	153
15.2.4 CONFIGURAR ENTRENAMIENTO	154
15.2.5 EXTRAER LOS DATOS Y GRÁFICAS.....	155
15.2.6 LANZAR ENTRENAMIENTO	156
15.2.7 GUARDAR ENTRENAMIENTO.....	156
15.2.8 ANALIZAR DATOS CON LA RED ENTRENADA.....	157



ÍNDICE FIGURAS

Figura 1.1-1: Requerimientos exigidos a los automóviles actuales.....	3
Figura 2.2.1-1: Airbags exteriores para proteger al peatón en caso de atropello.	14
Figura 2.2.1-2: Diseño frontal de los vehículos.	15
Figura 2.2.1-3: Sistema pop-up.	15
Figura 2.2.2-1: Comparación sistemas de frenado ABS y BAS.....	16
Figura 2.2.2-2: Funcionamiento amortiguadores magnetorreológicos.....	16
Figura 2.2.2-3: Sistema iluminación (Adaptative Forward Lighting)	17
Figura 2.2.2-4: Sistema visibilidad nocturna mediante cámaras infrarrojas.	18
Figura 2.2.2-5: Sistema detección de obstáculos.....	18
Figura 3.2-1: Diagrama de fuerzas de un vehículo tomando una curva.	22
Figura 4.1-1: Sistemas básicos del ESP	27
Figura 4.2-1: Sensores y componentes del ESP.	30
Figura 4.2-2: Mapa de flujo de información del ESP.....	30
Figura 4.2-3: Situación de Subviraje	31
Figura 4.2-4: Situación de Sobreviraje.....	31
Figura 5.1-1: Sistema coordenadas vehículo.....	37
Figura 5.2-1: Sistema coordenadas vehículo.....	38
Figura 5.2-2: Esquema de signo del ángulo de deriva.....	39
Figura 5.2-3: Sistema plano del vehículo.	39
Figura 5.2-4: Ángulo de deriva de un neumático.	40
Figura 5.3-2: Relación entre el ángulo de deriva de un neumático y las fuerzas laterales.	41
Figura 6.3-1: Esquema estudio [61].	50
Figura 6.3-2: Esquema estudio [33].	51
Figura 6.4-1: Esquema estudio [22].	53
Figura 7.1-1: Neurona biológica.....	59
Figura 7.3-1: Esquema de una RNA.....	61
Figura 7.4.2-1: Funciones de activación. A) Función lineal. B) Función signo. C) Función escalón. D) Función no lineal hiperbólica. E) Función no lineal tangente. F) Función no lineal sinusoidal.	62
Figura 7.4.3-1: Funcionamiento de una neurona de la red.	63
Figura 8.2-1: Estructura de una red.	75
Figura 8.2-2: Componentes de una neurona.	75
Figura 8.2-3: Reparto de pesos en la primera capa de neuronas.	76
Figura 8.2-4: Repartos de pesos en la segunda capa de neuronas.	76
Figura 8.2-5: Señal de salida.	76
Figura 8.2-6: Diferencia entre la salida obtenida y la salida deseada.....	77
Figura 8.2-7: Propagación del error en la capa intermedia.....	77
Figura 8.2-8: Propagación del error en la primera capa de la red.....	77
Figura 2.6.2-9: Modificación de los pesos en la capa de entrada.	77
Figura 2.6.2-103: Modificación de los pesos en la capa intermedia.	78
Figura 2.6.2-11: Propagación del error hasta la capa de salida.	78

Figura 9.2-1: CarSim.	81
Figura 9.3-1: Java-NNS.....	84
Figura 10.2.1.1-1: Test realizados en [33].....	91
Figura 10.2.1.2-1: Circuito para patrones de validación.	94
Figura 10.2.2.1-1: Vehículo clase D sedán utilizado en las maniobras en CarSim.	95
Figura 10.2.2.2-1: Menú configuración de la maniobra en CarSim.	96
Figura 10.2.2.2-2: Configuración del ángulo de dirección.....	96
Figura 10.2.2.2-3: Configuración del entorno.	97
Figura 10.2.2.3-1: Parámetros seleccionados.	98
Figura 10.2.2.3-2: Gráficas obtenidas de la simulación de la maniobra 1.....	98
Figura 10.2.2.3-3: Valores de la velocidad en la maniobra 1 para cada instante de tiempo.....	99
Figura 10.3-1: Encabezado del documento de patrones.	99
Figura 10.3-2: Patrones de entrenamiento (izda.) y validación (dcha.).....	101
Figura 10.4-1: Nomenclatura estructura redes.	102
Figura 10.4-2: Nomenclatura estructura redes. En rojo se define el tipo de capa, en azul la función de activación y en verde la función de salida de todas las neuronas de cada capa.....	103
Figura 10.4-4: Representación de cómo recorre la superficie de error al variar la tasa de aprendizaje.	104
Figura 10.4-4: Variación de la tasa de aprendizaje según la evolución del error.	104
Figura 10.4-5: Pantalla entrenamiento redes.	105
Figura 10.4-6: Pantalla validación de las redes.	105
Figura 10.5.1-1: Evolución del MSE de la Red 1: (4)-(5)-(1).....	106
Figura 10.5.2-1: Evolución del MSE de la Red 2: (4)-(10)-(1).....	108
Figura 10.5.3-1: Evolución del MSE de la Red 3: (4)-(15)-(1).....	109
Figura 10.5.4-1: Evolución del MSE de la Red 4: (4)-(20)-(1).....	111
Figura 10.5.5-1: Evolución del MSE de la Red 5: (4)-(25)-(1).....	113
Figura 10.5.6-1: Evolución del MSE de la Red 6: (4)-(5-5)-(1).....	114
Figura 10.5.7-1: Evolución del MSE de la Red 7: (4)-(10-10)-(1).....	116
Figura 10.6-1: Parámetros de la Red 2: (4)-(10)-(1)	119
Figura 15.1-1: Ventana principal del software CarSim.	145
Figura 15.1.1-1: Menú configuración vehículo.....	146
Figura 15.1.1-2: Submenú configuración vehículo.	146
Figura 15.1.2-1: Menú configuración maniobra y entorno.	147
Figura 15.1.2-2: Submenú configuración maniobra y entorno.....	148
Figura 15.1.2-3: Configuración de la dirección.....	149
Figura 15.1.2-4: Configuración del coeficiente de adherencia.....	149
Figura 15.1.4-1: Visualización de las variables de la dinámica del vehículo tras la simulación.....	149
Figura 15.2.1-1: Create layers	151
Figura 15.2.1-2: Creación de las neuronas de cada capa de una red.	152
Figura 15.2.1-3: Ventana editar neuronas	152
Figura 15.2.1-4: Ventana editar red.....	152
Figura 15.2.2-1: Crear conexiones de las neuronas.....	153
Figura 15.2.3-1: Documento de patrones.	153
Figura 15.2.4-1: Configuración entrenamiento: A) Initializing. B) Updating. C) Patterns. D) Learning	155



Figura 15.2.5-1: Ventana principal de simulación de una red.....	156
Figura 15.2.7-1: Guardar archivo .RES.....	156
Figura 15.2.8-1: Configuración panel de control para análisis de la red entrenada.	157
Figura 15.2.8-2: Pestaña setup para análisis de la red entrenada.	157
Figura 15.2.8-3: Pestaña test para análisis de la red entrenada.	158
Figura 15.2.8-4: Análisis de una red entrenada.....	158



ÍNDICE TABLAS

Tabla 2.1-1: Comparación de muertes de tráfico en los años 2001 y 2011.....	12
Tabla 3.2-1: Ecuaciones de velocidad de vuelco y derrape.....	23
Tabla 10.2.1.1-1: Test realizados en [70].....	91
Tabla 10.2.1.1-2: Test realizados en [59].....	92
Tabla 10.2.1.1-3: Test realizados en [36].....	92
Tabla 10.2.1.1-4: Test realizados en [68].....	93
Tabla 10.2.1.1-5: Valores que toma cada variable en las distintas maniobras.	93
Tabla 10.2.1.1-6: Maniobras simuladas.	94
Tabla 10.2.2-1: Parámetros maniobra 1.	95
Tabla 10.3-1: Valor β_{max} con el que se normaliza.....	100
Tabla 10.5.1-1: MSE Red 1: (4)-(5)-(1).....	106
Tabla 10.5.1-2: Errores. Red 1: (4)-(5)-(1).....	107
Tabla 10.5.2-1: MSE Red 2: (4)-(10)-(1).....	108
Tabla 10.5.2-2: Errores. Red 2: (4)-(10)-(1).....	108
Tabla 10.5.3-1: MSE Red 3: (4)-(15)-(1).....	109
Tabla 10.5.3-2: Errores. Red 3: (4)-(15)-(1).....	110
Tabla 10.5.4-1: MSE Red 4: (4)-(20)-(1).....	111
Tabla 10.5.4-2: Errores. Red 4: (4)-(20)-(1).....	111
Tabla 10.5.5-1: MSE Red 5: (4)-(25)-(1).....	113
Tabla 10.5.5-2: Errores. Red 5: (4)-(25)-(1).....	113
Tabla 10.5.6-1: MSE Red 6: (4)-(5-5)-(1).....	114
Tabla 10.5.6-2: Errores. Red 6: (4)-(5-5)-(1).....	115
Tabla 10.5.7-1: MSE Red 7: (4)-(10-10)-(1).....	116
Tabla 10.5.2-2: Errores. Red 2: (4)-(10)-(1).....	116
Tabla 10.6-1: Mínimos MSE para cada red.....	117
Tabla 10.6-2: Resumen errores.	118
Tabla 12.1-1: Errores normalizados y reales obtenidos para la red 2.....	128

ÍNDICE GRÁFICOS

Gráfico 2.1-1: Estadística muertes en accidentes de tráfico en España desde 2001 al 2011.....	11
Gráfico 10.5.1-1: Entrenamiento Red 1: (4)-(5)-(1).....	107
Gráfico 10.5.1-2: Validación Red 1: (4)-(5)-(1).....	107
Gráfico 10.5.2-1: Entrenamiento Red 2: (4)-(10)-(1).....	108
Gráfico 10.5.2-2: Validación Red 2: (4)-(10)-(1).....	109
Gráfico 10.5.3-1: Entrenamiento Red 3: (4)-(15)-(1).....	110
Gráfico 10.5.3-2: Validación Red 3: (4)-(15)-(1).....	110
Gráfico 10.5.4-1: Entrenamiento Red 4: (4)-(20)-(1).....	112
Gráfico 10.5.4-2: Validación Red 4: (4)-(20)-(1).....	112
Gráfico 10.5.5-1: Entrenamiento Red 5: (4)-(25)-(1).....	113
Gráfico 10.5.5-2: Validación Red 5: (4)-(25)-(1).....	114
Gráfico 10.5.6-1: Entrenamiento Red 6: (4)-(5-5)-(1)	115
Gráfico 10.5.6-2: Validación Red 6: (4)-(5-5)-(1)	115
Gráfico 10.5.7-1: Entrenamiento Red 7: (4)-(10-10)-(1)	116
Gráfico 10.5.7-2: Validación Red 7: (4)-(10-10)-(1)	117

1 INTRODUCCIÓN



1.1 INTRODUCCIÓN

En apenas un siglo, el vehículo automóvil se ha convertido, en el principal medio de transporte, en un producto muy relacionado con el desarrollo y riqueza de la sociedad y el instrumento más eficaz para dar satisfacción a los deseos de movilidad, autonomía y libertad del hombre actual.

El incremento exponencial del número de vehículos ha provocado problemas importantes, entre los que destacan: las enormes pérdidas de vidas humanas y materiales originadas por los accidentes; el gran impacto medioambiental que ocasionan, tanto atmosféricamente como acústicamente; y los problemas derivados de la congestión de tráfico en las grandes ciudades. Estos problemas han ido alterando a los ciudadanos y autoridades respecto al peligro que entraña el uso generalizado del automóvil y la respuesta ha sido un aumento notable, sobre todo en las dos últimas décadas, de exigencias técnicas contenidas en abundantes legislaciones nacionales e internacionales. Todas estas circunstancias han configurado un conjunto de requerimientos exigidos a los automóviles actuales y que se resumen en la Figura 1.1.-1

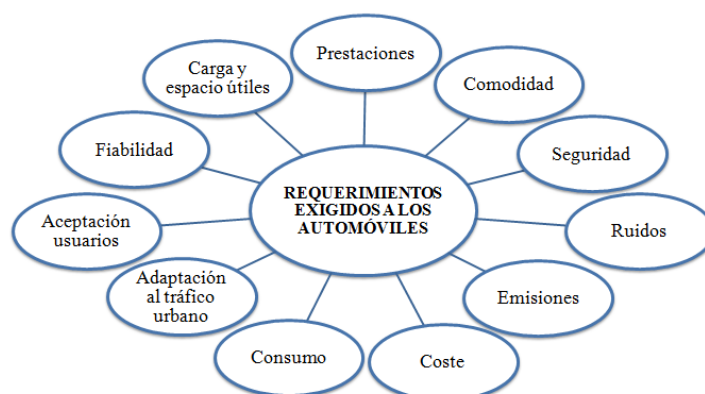


Figura 1.1-1: Requerimientos exigidos a los automóviles actuales.

Todos estos requerimientos no son independientes entre sí, sino que se encuentran prácticamente interrelacionados cada uno con todos los demás y existe un elevado número de conflictos entre ellos. El problema fundamental que afrontan los ingenieros, diseñadores y fabricantes es encontrar soluciones que constituyan buenos compromisos entre tan amplio y variado conjunto de requerimientos. [31]

Los vehículos actuales incorporan múltiples sistemas de seguridad para dar solución al primero de los problemas mencionados anteriormente, evitar accidentes de tráfico y/o disminuir los daños causados y, por lo tanto, reducir las pérdidas humanas y materiales. Para llevar a cabo el desarrollo de un sistema de seguridad es necesario extraer de la dinámica del automóvil una serie de parámetros que adecuadamente tratados y analizados permitan al vehículo, de manera externa al conductor, tomar decisiones y actuar. El problema radica en que existen parámetros que no son

fácilmente medibles o las herramientas necesarias para obtenerlos no constituyen un buen compromiso con el resto de requerimientos exigidos en los automóviles por parte de los usuarios/clientes o autoridades.

Ese es el caso del parámetro beta, ángulo de deriva del vehículo. Se trata de un parámetro fundamental para el desarrollo de nuevos y mejores sistemas control de estabilidad ya que es el ángulo formado entre la eje longitudinal del vehículo y hacia dónde se desplaza. Hasta no hace mucho no era posible medirlo directamente y sus estimaciones no eran del todo precisas pero estudios recientes han conseguido hallarlo aunque el coste asociado no permite instalarlo en los vehículos de producción por lo que es necesario seguir investigando nuevas maneras de obtenerlo, reduciendo drásticamente su coste sin que se vea disminuido su exactitud ni precisión.

El presente proyecto consiste en dar solución a la problemática actual, enfocando un nuevo método de estimación del ángulo de deriva mediante redes neuronales. Para ello, se extraerán los parámetros necesarios del software de simulación CarSim y posteriormente serán introducidos en el software Java-NNS que permite crear las redes neuronales y realizar el aprendizaje y validación de las mismas.

1.2 OBJETIVOS DEL PROYECTO

Los objetivos que se pretenden cumplir en el presente Proyecto Fin de Carrera son los siguientes:

- ❖ Tratar, de forma breve, aquellos aspectos más importantes e interesantes que están relacionados con la dinámica lateral del vehículo, así como los sistemas de seguridad incorporados en los vehículos para negociar con las inestabilidades que se puedan producir en el automóvil durante su marcha.
- ❖ Profundizar en el conocimiento del ángulo de deriva de un automóvil, uno de los parámetros claves en la dinámica lateral.
- ❖ Realizar un repaso sobre las técnicas y estrategias seguidas por diferentes autores en sus estudios con el objetivo de medir o estimar el ángulo de deriva. En especial, se profundizará sobre las Redes Neuronales Artificiales.
- ❖ Estimación del ángulo de deriva de un vehículo mediante la técnica de Redes Neuronales Artificiales, haciendo uso del Software Java-NNS, introduciendo como parámetros de entrada al sistema: la velocidad longitudinal, la aceleración lateral, la velocidad de guiñada y el ángulo de dirección obtenidos del software CarSim.



1.3 ESTRUCTURA DEL PROYECTO

El proyecto está estructurado en diversos capítulos y varios anexos, que son descritos a continuación:

En el Capítulo 1 se presenta la introducción al trabajo realizado así como los objetivos marcados.

En el Capítulo 2 se hace un breve repaso a los sistemas de seguridad que hoy en día se incorporan a los vehículos de producción y la importancia que tienen en la disminución de accidentes tráfico y, por ende, de víctimas en carretera.

En el Capítulo 3 se realiza un acercamiento a la dinámica lateral de los vehículos para conocer las inestabilidades que se pueden presentar cuando un vehículo realiza un giro ya que se puede perder la adherencia y, por consiguiente, el control del vehículo o incluso volcar.

En el Capítulo 4 se profundiza en el sistema de control de estabilidad (ESP) ideado a causa de la problemática existente relativa a la dinámica lateral de los vehículos.

El Capítulo 5 está dedicado al ángulo de deriva de un vehículo, (uno de los parámetros clave en la dinámica de los automóviles y cuyo conocimiento es vital para el desarrollo y mejora de los sistemas de seguridad en los vehículos.

En el Capítulo 6 se hace un repaso a los estudios previos que se encuentran en la literatura sobre el ángulo de deriva de un vehículo con el fin de conocer las distintas estrategias que se han seguido para su medición o estimación.

El Capítulo 7 se centra en la técnica de redes neuronales artificiales pues va a ser la estrategia seguida en este estudio para estimar el ángulo de deriva.

En el Capítulo 8 se introduce el tipo de aprendizaje que se va llevar cabo en las redes neuronales.

En el Capítulo 9 se presentan los Softwares que se van a utilizar para la estimación del ángulo de deriva de un vehículo. Se usa el programa de simulación de la dinámica vehicular, CarSim, para obtener los datos necesarios que servirán para el entrenamiento y validación de las redes neuronales definidas y simuladas en el programa Java-NNS.

En el Capítulo 10 está dedicado a la parte experimental de este Proyecto, el cual consiste en la estimación del ángulo de deriva de un vehículo mediante la simulación de diferentes

configuraciones de redes neuronales entrenadas y validadas con patrones obtenidos de diferentes maniobras simuladas en CarSim.

En el Capítulo 11 se realiza el presupuesto del Proyecto.

En el Capítulo 12 se expresan las conclusiones finales que se han alcanzado tras el análisis de los resultados obtenidos y los posibles trabajos futuros que no se han podido llevar a cabo por falta de tiempo y/o medios, pero que se consideran importantes para avanzar en la investigación, desarrollo y mejora de los sistemas de seguridad implementados en los vehículos de producción.

En el Capítulo 13 se muestra la bibliografía consultada.

El Capítulo 14, Anexo I, donde se ha dispuesto la tabla sobre la estadística de accidentes de tráfico de la que se han extraído los datos mostrados en el Capítulo 2.

El Capítulo 15, Anexo II, donde se han incluido unos breves tutoriales básicos sobre los Softwares, CarSim y Java-NNS, utilizados en este estudio.

2 SISTEMAS DE SEGURIDAD EN LOS VEHÍCULOS



2.1 INTRODUCCIÓN

Según la Organización Mundial de la Salud 1 (OMS) 800.000 personas mueren cada año en el mundo por culpa de accidentes de carretera y alrededor de 20 millones resultan heridas. En Europa, cada año, 65.000 vidas se pierden para siempre sobre el asfalto. En particular, en España, 9.000 personas fallecen anualmente en los treinta días siguientes de sufrir un accidente, 15.000 quedan permanentemente inválidas y 150.000 resultan heridas de consideración [62].

A la vista de esas cifras terroríficas es necesario tomar medidas inminentemente. Es por ello que los fabricantes y estados invierten grandes esfuerzos y millones en desarrollar y mejorar sistemas de seguridad en los vehículos.

Según el informe de siniestralidad de 2013 presentado por la Directora General de Tráfico, María Seguí Gómez [25], se confirma que esos esfuerzos están dando sus frutos. Desde 1998 se han ido reduciendo de manera sostenible el número muertos en accidentes producidos en las vías españolas, Gráfico 2.1-1. Como se puede observar, se ha reducido la siniestralidad en más de la mitad en trayectos interurbanos, pasando de 4543 accidentes en 2001 a 1603 en 2011. Del mismo modo ocurre en trayectos urbanos, donde se ha registrado 457 accidentes en 2011 frente a los 974 en 2001. Aunque lo logrado es motivo de alegría y optimismo, es necesario seguir luchando y no conformarse, pues sigue siendo el número de fallecidos demasiado elevado.

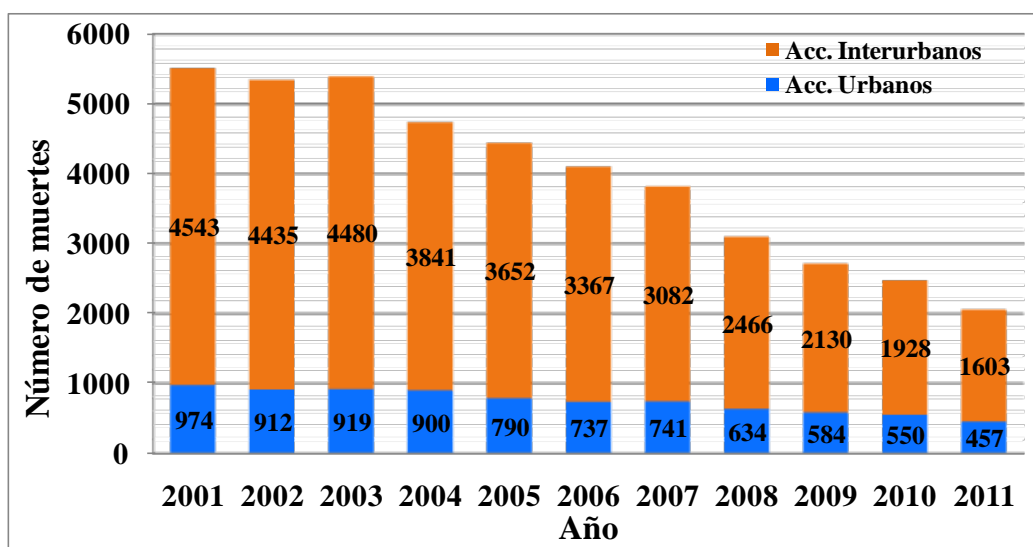


Gráfico 2.1-1: Estadística muertes en accidentes de tráfico en España desde 2001 al 2011.

En el Anexo I, Capítulo 14, de este documento se ha adjuntado una tabla resumen en el que se puede consultar el número de muertos por accidentes de tráfico desde 2001 al 2011 en vías españolas y la causa atribuida a cada uno de ellos, distinguiendo entre zonas urbanas e interurbanas [25].

A continuación se realizará una comparación entre los datos de los años 2001 y 2011, una década de diferencia. En la Tabla 2.1-1 se muestran los valores dependiendo de la causa y de la zona donde se produjo, así como el total. Además, se completa con el porcentaje que supone respecto al número de fallecidos en ese año.

Tabla 2.1-1: Comparación de muertes de tráfico en los años 2001 y 2011.

Muertes Tráfico 2001 y 2011		URBANA		INTERURBANA		TOTAL	
Causa	Año	Muertes	%	Muertes	%	Muertes	%
Salida de la vía	2001	164	17%	1737	38%	1901	34%
	2011	68	15%	578	36%	646	31%
Colisión frontal	2001	61	6%	772	17%	803	15%
	2011	14	3%	322	20%	336	16%
Colisión latera y frontolateral	2001	215	22%	958	21%	1173	21%
	2011	75	16%	254	16%	329	16%
Colisión trasera y múltiple	2001	50	5%	332	7%	382	7%
	2011	28	6%	163	10%	191	9%
Atropello a peatón	2001	372	38%	445	10%	817	15%
	2011	217	47%	150	9%	367	18%
Vuelco	2001	14	1%	83	2%	97	2%
	2011	9	2%	38	2%	47	2%
Otro tipo de accidente	2001	98	10%	216	5%	314	6%
	2011	46	10%	98	6%	144	7%
Total	2001	974	100%	4543	100%	5517	100%
	2011	457	100%	1603	100%	2060	100%

Tras analizar los datos se pueden obtener las siguientes conclusiones de interés:

- ❖ En 2011 se ha visto reducido de manera general la mortalidad respecto al año 2001 en más de un 50%.
- ❖ Salvo en atropellos de peatón en zona urbana que se ha visto incrementado el porcentaje de fallecidos considerablemente, el resto de porcentajes se mantienen prácticamente constantes. Esto indica que se ha trabajado de manera homogénea en la reducción de la siniestralidad en España.
- ❖ La salida de la vía es la causa de mayor número de muertos en España con un 36%. Es en vías interurbanas donde se registra mayor número.
- ❖ La segunda causa de muertes es la de un peatón atropellado llegando a ser el 18% del total de muertes. La mayoría de estas se producen en zonas urbanas.

Es necesario tomar consciencia e intentar dar solución al número de muertes de tráfico, en especial las provocadas por un atropello o una salida de la vía. En el siguiente apartado se comentará los sistemas de seguridad que los vehículos incorporan para hacer que cada vez haya menos fallecidos en las carreteras de España. En especial se hará mayor énfasis en los sistemas de estabilidad lateral ya que un estudio, [61], revela que el 27% de los accidentes son debidos a una pérdida de control del vehículo y, además, están íntimamente relacionados con el objetivo de este proyecto.

2.2 SISTEMAS DE SEGURIDAD

Aunque desde los años 50, momento en el que se comenzaron a instalar en los vehículos los cinturones de seguridad, no ha sido hasta la década de 1990 cuando se ha tomado consciencia seriamente de la problemática que se tenía y el número de personas que perdían la vida por un accidente de tráfico. Desde los 90 se han ido introduciendo sistemas de seguridad en los vehículos con el objetivo de evitar accidentes o disminuir los daños causados a los ocupantes o a terceras personas que se puedan ver involucradas. Hoy en día, tanto por parte de los gobiernos como de los fabricantes se está potenciando la implantación de mejores y nuevos sistemas de seguridad. Pueden ser clasificados según su función. Hay sistemas de seguridad activos y pasivos: [11], [34], [65], [69].

2.2.1 SEGURIDAD PASIVA

Elementos que reducen al mínimo los daños que se pueden producir cuando el accidente es inevitable:

A) Los cinturones de seguridad

Imprescindibles para cualquier viajero, básicos en la seguridad vial. En caso de impacto, cuentan con un dispositivo que bloquea el mecanismo en caso de sufrir una fuerte desaceleración. Evitan que la persona salga despedida. Además, en caso de accidente, cuentan con hebillas de apertura fácil.

B) Reposacabezas

Son los elementos fundamentales en la protección de la persona frente al latigazo cervical. Sistemas actuales permiten acompañar a la cabeza de la persona durante la colisión y disminuir las lesiones.

C) Los Airbags

Son unas bolsas que, mediante un sistema pirotécnico, se inflan en fracciones de segundo cuando el coche choca con un objeto sólido a una velocidad considerable. Su objetivo es impedir que los ocupantes se golpeen directamente con alguna parte del vehículo. Actualmente existen las bolsas frontales, laterales, tipo cortina (para la cabeza) e incluso para las rodillas.

Complementando a los airbags situados en el interior del vehículo, se están desarrollando nuevos modelos de airbags delanteros exteriores para ofrecer protección a los peatones en caso de atropello. El sistema de activación de los airbags del parachoques se basa un sistema de detección de obstáculos, o peatones, ya que debe inflarse antes de que se produzca el contacto.



Figura 2.2.1-1: Airbags exteriores para proteger al peatón en caso de atropello.

D) Sistemas anti-incendios

Diseño del depósito de combustible y elementos auxiliares para evitar el derrame de combustible en caso de colisión y cierre automático de la inyección de combustible para impedir incendios.

E) Aviso automático a centro de emergencias después de un accidente

F) Puertas diseñadas para una fácil apertura después del accidente

G) Diseño del frontal

Ha cambiado en los últimos años para que en caso de accidente, la máxima energía posible sea absorbida por el vehículo mediante la deformación de la estructura. Además, se buscan formas más suavizadas, con menos aristas; parachoques y capós menos rígidos; que cumplan los parámetros definidos por EEVC/WG17, con el objetivo de que las lesiones producidas en el peatón en caso de atropello se vean reducidas.

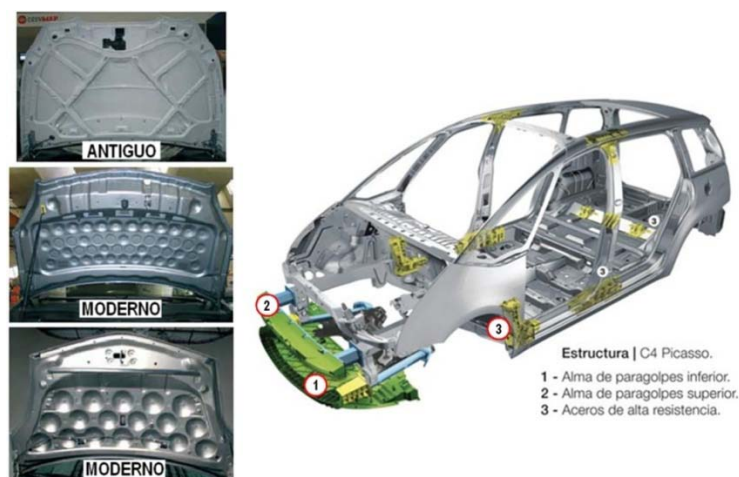


Figura 2.2.1-2: Diseño frontal de los vehículos.

H) Sistema pop-up

Diseñado para elevar la parte del capó más alejada del frontal a una determinada altura, amortiguando el impacto entre el peatón y el vehículo, reduciendo considerablemente las lesiones ocasionadas a los peatones. Esto es posible ya que genera un espacio libre entre el capó y las partes duras del vehículo, como pueden ser el motor u otros elementos estructurales.

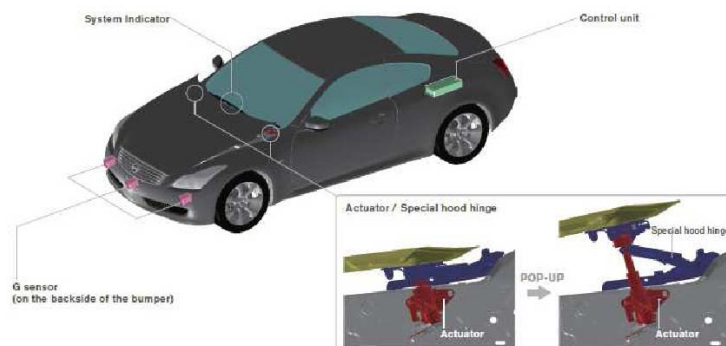


Figura 2.2.1-3: Sistema pop-up.

2.2.2 SEGURIDAD ACTIVA

Es el conjunto de todos aquellos elementos que contribuyen a proporcionar una mayor eficacia y estabilidad al vehículo en marcha, y en la medida de lo posible, evitar un accidente. Entre ellos cabe destacar:

A) Sistema de frenado

Todos los sistemas de frenado actuales cuentan con circuitos independientes que permiten frenar con seguridad en caso de que alguno falle.

Además, dependiendo del modo de funcionamiento y de su utilidad se pueden clasificar en tres subgrupos:

- ❖ Autónomos: Tratan de evitar o reducir el riesgo de accidente percibido.
- ❖ Emergencia: Interviene sólo en situaciones críticas.
- ❖ De frenado: Acciona los frenos, directamente.

El ABS (Antilock Braking System) trabaja regulando la frenada para evitar el bloqueo de las ruedas y manteniendo la capacidad de cambiar de dirección para evadir obstáculos. Desde su aparición se ha ido perfeccionando dando lugar a nuevos modelos aún más seguros: el asistente de frenada de emergencia BAS (Brake Assist System), el repartidor de frenada electrónico EBV (EBD: Electronic Brake Variation System) o los frenos direccionales SERVOTRONIC.

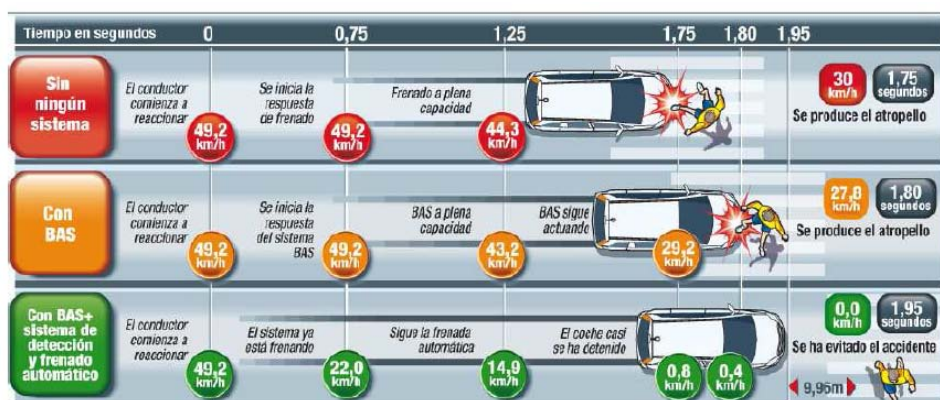


Figura 2.2.2-1: Comparación sistemas de frenado ABS y BAS.

B) Sistema de suspensión

La suspensión es un componente fundamental en los vehículos. Como funciones principales están la de absorber las irregularidades del terreno, el guiado correcto de la rueda sobre el terreno tanto en términos angulares como de traslación y la maximización de la adherencia de la rueda sobre un firme irregular. Es por ello, que el sistema de suspensión juega un papel muy importante en la seguridad y es necesario incorporar nuevos sistemas que permitan aumentar su efectividad y su respuesta ante situaciones adversas. Expertos investigan suspensiones activas como pueden ser las magnetorreológicas que a través de campos magnéticos consiguen variar el comportamiento del fluido que hay en el interior de los amortiguadores y, por tanto, su modo de funcionamiento.

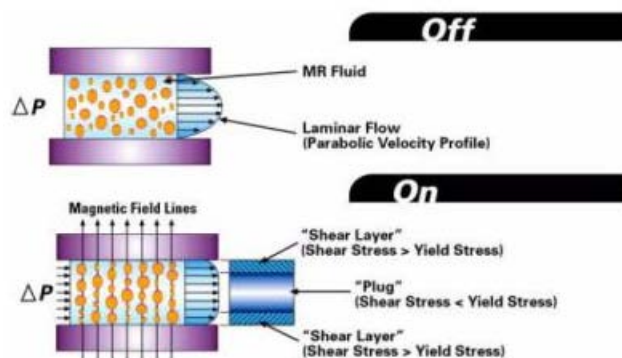


Figura 2.2.2-2: Funcionamiento amortiguadores magnetorreológicos.

C) Sistemas de iluminación

El uso de las luces es un punto fundamental dentro de la seguridad activa ya que gracias a las mismas, los conductores y usuarios de la vía pública se comunican, es tan importante ver bien como ser vistos. Es por ello, que los sistemas de iluminación han evolucionado desde hace unos años ya que los faros eran muy débiles y las luces no eran blancas.

Se puede destacar el sistema “Adaptative Forward Lighting” de Opel. Activa las luces ante situaciones inesperadas incrementando la visibilidad en condiciones de poca luminosidad como curvas, intersecciones o cuando la climatología dificulta la visibilidad.

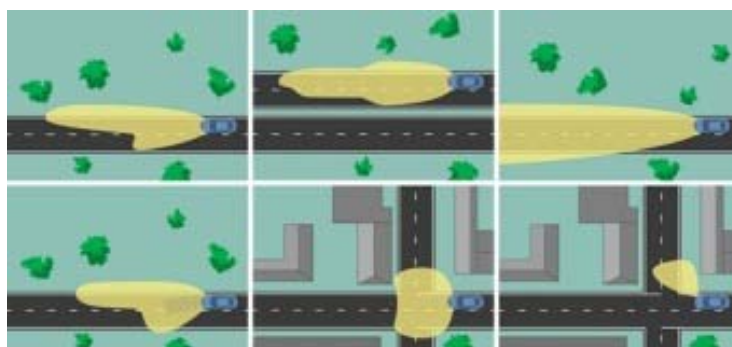


Figura 2.2.2-3: Sistema iluminación (Adaptative Forward Lighting)

D) El sistema de dirección

Garantiza la correcta maniobra del vehículo. Los sistemas de dirección de los coches actuales se endurecen a altas velocidades para evitar posibles accidentes.

E) Sistemas de aviso

Existen diversos sistemas cuya función es la de avisar al conductor. Cabe destacar los siguientes:

No te pases de la raya:

Avisa al conductor cuando el coche pisa la línea de cambio de carril y no se activan los intermitentes para indicarlo. Existen dos tipos:

- Aviso de cambio de carril: Señal acústica o vibratoria del volante o el asiento del coche, cuando se desvía de su trayectoria.
- Permanencia en el carril: Frena las ruedas o mueve el volante para que el coche se mantenga en el centro del carril.

Para no dormirse al volante:

Sensores en el volante y algoritmos basados en la frecuencia de movimientos del conductor detectan si está fatigado. En caso de estarlo es alertado mediante señales acústicas o luminosas en el panel de mandos.

Visibilidad:

Todos aquellos sistemas que permiten al conductor conocer el entorno por el que circula. Algunos ejemplos son: eliminación de los ángulos muertos, ayudas de control como radares, comunicación de seguridad inalámbrica, visión nocturna, cámara infrarroja, cámara trasera...



Figura 2.2.2-4: Sistema visibilidad nocturna mediante cámaras infrarrojas.

F) Sistemas de detección de obstáculos

Se trata de sistemas basados en la tecnología de visión artificial. Son capaces de reconocer la naturaleza de diferentes objetos que pueden situarse delante del vehículo, incluidos los peatones. Mediante un sistema radar el vehículo es capaz de detectar y evaluar la amenaza que puede suponer el objeto. Estos sistemas pueden actuar de forma automática activando los frenos, consiguiendo una frenada más rápida y eficaz.



Figura 2.2.2-5: Sistema detección de obstáculos.

G) Sistemas de control de estabilidad

Actúan evitando que se pierda el control del vehículo cuando el vehículo está tomando una curva y el límite de adherencia del neumático o el momento de vuelco es superado. En el Capítulo 4 se profundizará más sobre estos sistemas. Las siglas más extendidas son ESP (programa electrónico de estabilidad)

3 DINÁMICA LATERAL



3.1 INTRODUCCIÓN

Antes de comenzar a explicar el modo de funcionamiento del sistema ESP se va a hacer un pequeño apunte a la dinámica lateral de un vehículo [31], [78].

Puesto que las maniobras que conllevan mayor riesgo son aquellas en las que el vehículo no está circulando en línea recta sino que está tomando una curva, realizando un adelantamiento, etc. es necesario prestarlas mayor atención y analizarlas con detenimiento.

Los factores que afectan a la dinámica lateral de un automóvil son las descritas a continuación:

- ❖ Dimensionales: Distancia entre ejes, vía.
- ❖ Reparto de masas: Posición del centro de gravedad, relación masa suspendida/ no suspendida.
- ❖ Aerodinámicas: coeficiente aerodinámicos de fuerza lateral y momento de guiñada.
- ❖ Neumáticos: Características laterales (rigidez de deriva y su variación en función de la carga) y esfuerzos transversales.
- ❖ Suspensión: características geométricas y dinámicas.
- ❖ Factores externos o características del medio: el ángulo de incidencia del aire, las irregularidades de la calzada, el radio de curva y peralte de la calzada y el coeficiente de adherencia.
- ❖ Velocidad: es un factor importante ya que puede existir un valor crítico de la misma a partir del cual el vehículo presenta un comportamiento inestable.

3.2 CIRCULACIÓN EN CURVA

Cuando un vehículo se encuentra circulando en una curva a velocidad apreciable aparece la aceleración centrífuga que debe poder ser compensada con la aparición de las fuerzas en la interfase neumático-calzada o con la aparición del momento de vuelco. Aún así, existirá una velocidad a partir de la cual se perderá la adherencia con la superficie produciéndose el derrape del vehículo o el vuelco del automóvil debido a que el momento tomará valores muy elevados. A continuación se estudiarán por separado ambos casos.

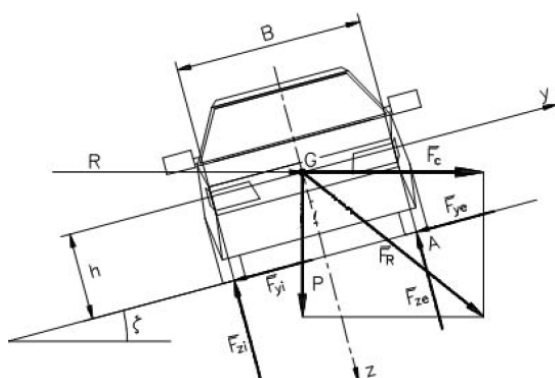


Figura 3.2-1: Diagrama de fuerzas de un vehículo tomando una curva.

Del diagrama de fuerzas representado en la figura 3.2-1, cuando un vehículo está tomando una curva con peralte, se pueden obtener las fuerzas normales a la superficie de rodadura - ecuación 3.2-1 - y las fuerzas en dirección “y” - ecuación 3.2-2 -.

$$F_z = F_{zi} + F_{ze} = P \cdot \cos \xi + F_c \cdot \sin \xi \quad \text{Ec. 3.2-1}$$

$$F_y = F_{yi} + F_{ye} = -P \cdot \sin \xi + F_c \cdot \cos \xi \quad \text{Ec. 3.2-2}$$

3.2.1 VELOCIDAD LIMITE DE DERRAPE

Partiendo de las ecuaciones 3.2-1 y 3.2-2 e introduciéndolas en la siguiente ecuación que las relaciona se obtiene:

$$(F_{zi} + F_{ze}) \cdot \mu_y = F_{yi} + F_{ye} \longrightarrow (P \cdot \cos \xi + F_c \cdot \sin \xi) \cdot \mu_y = -P \cdot \sin \xi + F_c \cdot \cos \xi \quad \text{Ec. 3.2.1-1}$$

Además, conociendo la fuerza centrífuga:

$$F_c = \frac{m \cdot V^2}{R} \longrightarrow F_c = \frac{P \cdot V^2}{g \cdot R} \quad \text{Ec. 3.2.1-2}$$

Introduciendo Ec. 3.2.1-2 en Ec. 3.2.1-1 se obtiene la velocidad límite de derrape. Si la curva tiene peralte viene definida por la ecuación 3.2.1-3, por el contrario, será la ecuación 3.2.1-4:

$$V_{ld} = \sqrt{\frac{g \cdot R \cdot (\mu_{y \max} + \tan \xi)}{1 - \mu_{y \max} \cdot \tan \xi}} \quad \text{Ec. 3.2.1-3}$$

$$V_{ld}(\xi = 0) = \sqrt{g \cdot R \cdot \mu_{y \max}} \quad \text{Ec. 3.2.1-4}$$

Como se puede observar, la velocidad de derrape de un vehículo no es dependiente de ninguno de los parámetros del vehículo. Únicamente depende del coeficiente de adherencia entre el neumático y la calzada, el ángulo de peralte, si lo hubiera, y el radio de la curva.

3.2.2 VELOCIDAD LIMITE DE VUELCO

Partiendo de las ecuaciones 3.2-1 y 3.2-2 e introduciéndolas en la siguiente ecuación que las relaciona se obtiene:

$$\frac{F_y}{F_z} = \frac{-P \cdot \sin \xi + F_c \cdot \cos \xi}{P \cdot \cos \xi + F_c \cdot \sin \xi} \approx \frac{B}{2 \cdot h} \quad \text{Ec. 3.2.2-1}$$

Además, conociendo la fuerza centrífuga:

$$F_c = \frac{P \cdot V^2}{g \cdot R} \quad \text{Ec. 3.2.2-2}$$

Introduciendo Ec. 3.2.2-2 en Ec. 3.2.2-1 y realizando unas operaciones matemáticas se obtiene la velocidad límite de vuelco. Si la curva tiene peralte viene definida por la ecuación Ec. 3.2.2-3, por el contrario, será la ecuación Ec. 3.2.2-4:

$$V_{lv} = \sqrt{\frac{g \cdot R \cdot \left(\frac{B}{2 \cdot h} + \tan \xi \right)}{1 - \frac{B}{2 \cdot h} \cdot \tan \xi}} \quad \text{Ec. 3.2.2-3}$$

$$V_{lv}(\xi = 0) = \sqrt{g \cdot R \cdot \frac{B}{2 \cdot h}} \quad \text{Ec. 3.2.2-4}$$

En este caso, la velocidad de vuelco de un vehículo sí depende de algunos parámetros de un vehículo, en concreto, de la batalla y de la altura del centro de gravedad. Además, al igual que antes, también depende del coeficiente de adherencia, del ángulo de peralte y el radio de la curva.

En la Tabla 3.2-1 quedan recogidas las expresiones generales para la condición de vuelco y la de derrape:

Tabla 3.2-1: Ecuaciones de velocidad de vuelco y derrape.

VELOCIDAD VUELCO	VELOCIDAD DERRAPE
$V_{ld} = \sqrt{\frac{g \cdot R \cdot (\mu_{y \max} + \tan \xi)}{1 - \mu_{y \max} \cdot \tan \xi}}$	$V_{ld} = \sqrt{\frac{g \cdot R \cdot \left(\frac{B}{2 \cdot h} + \tan \xi \right)}{1 - \frac{B}{2 \cdot h} \cdot \tan \xi}}$

Si se observan las expresiones obtenidas para cada una de las velocidades anteriores, se ve claramente que se puede pasar de una a otra sin más que cambiar μ_{\max} por $\frac{B}{2 \cdot h}$ y viceversa, por tanto, se puede comparar ambos límites de la siguiente forma:

- ❖ Si $\mu_{y\max} = \frac{B}{2 \cdot h}$ ambas velocidades serán iguales por lo que el fenómeno del vuelco y el de derrape ocurrirán de forma simultánea.
- ❖ Si $\mu_{y\max} > \frac{B}{2 \cdot h}$ debido a que la velocidad de vuelco es menor y se alcanza antes que la de derrape se produciría el vuelco del vehículo sin llegar a derrapar. Este fenómeno se da en vehículos que tengan el centro de gravedad a una altura elevada cuando la adherencia de la vía sea suficiente. Normalmente estas características las cumplen los vehículos industriales circulando sobre pavimento seco.
- ❖ Si $\mu_{y\max} < \frac{B}{2 \cdot h}$ caso contrario al anterior, la velocidad de derrape se alcanza antes que la de vuelco por lo que el vehículo derraparía pero no llegaría a volcar. Este fenómeno se suele dar en turismos y vehículos industriales cuando la adherencia no es suficiente.

4 SISTEMA ESP



4.1 INTRODUCCIÓN

En contra de lo que podamos pensar la inmensa mayoría de los conductores, carecemos de las aptitudes y conocimientos necesarios para dominar nuestro automóvil cuando las cosas no están bajo control. En muchas ocasiones, es el conductor el que provoca el “problema”, y lo peor de todo es que lo hace sin saber. Es importante hacer hincapié en que el control de estabilidad no puede obrar milagros, no proporciona más agarre al coche; básicamente, lo que hace es enmendar los errores que el conductor comete o solucionar algunos imprevistos que no haya sido capaz de anticipar.

Es por ello, que la marca alemana Bosch, quien tiene la patente de este sistema revolucionario, comenzó a trabajar en un sistema de seguridad para mejorar la estabilidad en frenadas de emergencia en 1983. Fue en 1992 cuando se inició la producción del ESP 5.0. Este sistema incorporaba a las funciones del sistema ABS, creado en 1978, y el Control de Tracción, creado en 1986, los beneficios adicionales del control de estabilidad.

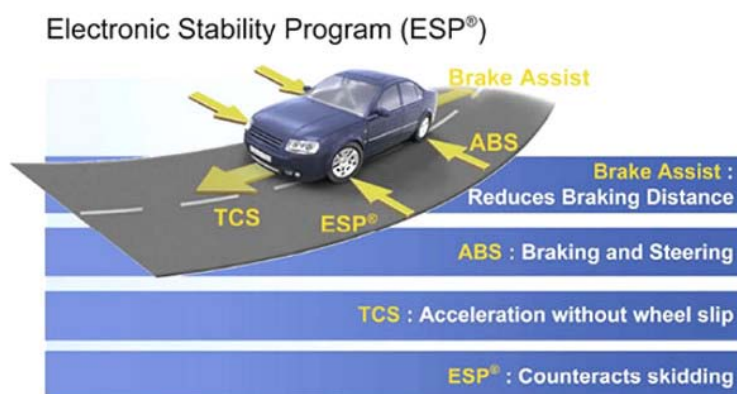


Figura 4.1-1: Sistemas básicos del ESP

Las siglas ESP hacen referencia a "Elektronisches Stabilitäts-Programm" (programa electrónico de estabilidad), un sistema de seguridad activo como ya se comentó en el apartado 2.2.2. Actualmente, recibe numerosas siglas, siendo la más común ESP, aunque hay algunos fabricantes que utilizan otras como ESC, VSC, VDC, DSC, etc., si bien el funcionamiento básico, sus componentes y su efectividad, es idéntico.

En 1995, la compañía Mercedes-Benz introdujo el sistema ESP en la gama alta de sus vehículos, SL600 y S600 pero el origen de que el sistema ESP se hiciera popular es debido a un gran chasco, con repercusión mundial, que sufrió la compañía Mercedes-Benz en una de sus pruebas de estabilidad en 1997. La prueba consistía en realizar un giro a la izquierda y otro a la derecha en muy poco tiempo a gran velocidad. Esta prueba, realizada con un modelo de la clase A, fue todo un desastre al provocarse el vuelco del vehículo. El piloto responsable fue un periodista

sueco, Robert Collin, que participó en la llamada prueba del Alce; nombre atribuido por el movimiento que hay que realizar para esquivar a este mamífero si aparece en la carretera.

Gracias a los avances en la electrónica se ha conseguido desarrollar y abaratar la producción de sensores que pueden recibir y transmitir datos de forma muy rápida y unidades de control capaces de tomar decisiones instantáneas y, sobre todo, interactuar entre ellas. Este último punto es crucial, ya que para que el ESP pueda variar la trayectoria de nuestro coche, es necesario que trabajen al unísono varios sistemas: el motor, los frenos y el ABS, la dirección...

Según estudios recientes, se calcula que este sistema podría evitar en torno a 600 muertes al año en España y ha sido considerado en Estados Unidos como el sistema de seguridad, tras el cinturón, con el mayor potencial para salvar vidas en accidentes de tráfico. Es por ello que a partir de 2011, todos los vehículos nuevos que se vendan en Europa deben incorporar obligatoriamente este sistema de control de la estabilidad [12], [13], [60], [71].

4.2 FUNCIONAMIENTO

El ESP es un sistema de seguridad activo. Su principal tarea es que el vehículo trace la trayectoria que el conductor le marque con el volante y, así, evitar posibles pérdidas de control del vehículo producidas por derrapes y evitando, de esta manera, posibles accidentes de tráfico. Para ello, el ESP debe ser capaz de dar respuesta a las siguientes dos cuestiones:

- ❖ Primera pregunta: ¿Hacia dónde conduce el conductor?
- ❖ Segunda pregunta: ¿Hacia dónde se dirige el vehículo?

Un microordenador instalado en el automóvil se encarga de dar respuesta a ambas preguntas chequeando 25 veces por segundo las señales provenientes de los sensores del ESP. En caso de no coincidir la dirección a la que se dirige el vehículo con la marcada por el conductor, el ESP actúa de inmediato y de manera independiente al conductor. El modo de corregir la dirección es frenando la rueda que más convenga en cada situación para generar una fuerza opuesta a la que impide que el vehículo trace la trayectoria fijada y la compensa hasta que se vuelve a la trayectoria ideal. Si la inercia es lo suficientemente elevada o la adherencia demasiado escasa, no podrá solucionar el problema. Además, simultáneamente, el ESP, actúa sobre el régimen de vueltas del motor y sobre el cambio de velocidades si es automático. Incluso, como la centralita también recibe información sobre la velocidad, puede actuar sobre la inyección, cortando el flujo de combustible y evitando que el conductor pueda aumentar la velocidad. Cuando el sistema de estabilidad está actuando, se encenderá un chivato en el cuadro de mandos.

A continuación, se explica cómo el ESP es capaz de dar respuesta a las dos cuestiones planteadas anteriormente.

- ❖ A la primera pregunta: El sistema recibe la respuesta del sensor goniométrico de la dirección, por el que se conocerá el ángulo que se ha girado el volante y por tanto hacia dónde el conductor quiere dirigirse.
- ❖ A la segunda pregunta: En este caso, la respuesta no se consigue de manera tan sencilla. El problema es que no es posible medirlo directamente debido al elevadísimo coste que supone por lo que se obtiene estimándolo a partir de otras magnitudes medidas en el vehículo que son: la aceleración lateral, la velocidad de guiñada y la velocidad longitudinal, hallada a su vez a partir de la velocidad angular de las ruedas.

Por tanto, los parámetros a los que se hace referencia y que son fundamentales para el correcto funcionamiento del sistema de estabilidad son los siguientes:

- ❖ **Ángulo de dirección (δ)**
- ❖ **Velocidad angular de las ruedas (ω)**
- ❖ **Aceleración lateral (a_y)**
- ❖ **Velocidad de guiñada ($\dot{\psi}$)**

Para obtener dichas señales y que el ESP pueda actuar es necesario instalar en el vehículo una serie de sensores y componentes:

- ❖ **Sensor de ángulo de volante:** mediante este sensor, el ESP conoce la posición del volante. No sólo indica cuánto se ha girado, sino también cómo de rápido se ha hecho.
- ❖ **Sensores de giro de rueda:** son los mismos del ABS y su misión es informar a la unidad de control sobre el comportamiento de las ruedas a través de su velocidad de rotación.
- ❖ **Sensor de posición del acelerador:** Indica cuánto se tiene presionado el acelerador.
- ❖ **Sensores de aceleración lateral:** indican el comportamiento real del vehículo, si está describiendo una curva o no. Si el sensor del volante registra que el volante está girado pero no hay una aceleración lateral, significa que el coche no está describiendo la trayectoria deseada.
- ❖ **Un giróscopo:** este sensor indica la velocidad de guiñada, si el vehículo está girando sobre su propio eje.
- ❖ **Unidad de control:** Se trata de un ordenador que compara los datos que recibe de los sensores con los datos que debería tener. Si no coinciden, el ESP se activa.

- ❖ **Grupo hidráulico:** ejecuta las órdenes de la unidad de control y regula, mediante las electroválvulas, la presión de los cilindros. La unidad de control se encarga del control eléctrico y electrónico así como de todas las funciones de regulación del sistema.

En las siguientes figuras se muestran los componentes y el lugar habitual donde se instalan, así como un mapa del flujo de información de los sensores a la centralita y de esta a los actuadores.

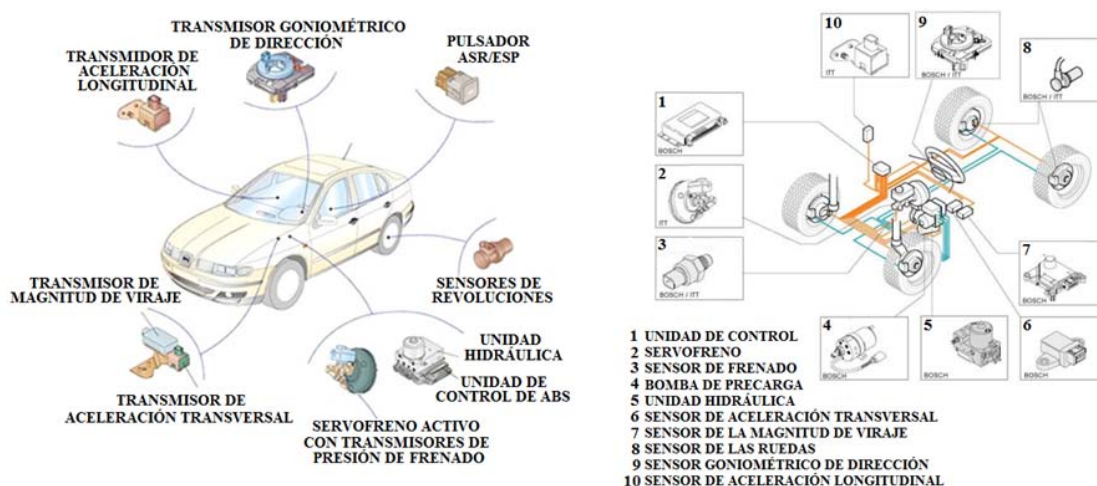


Figura 4.2-1: Sensores y componentes del ESP.

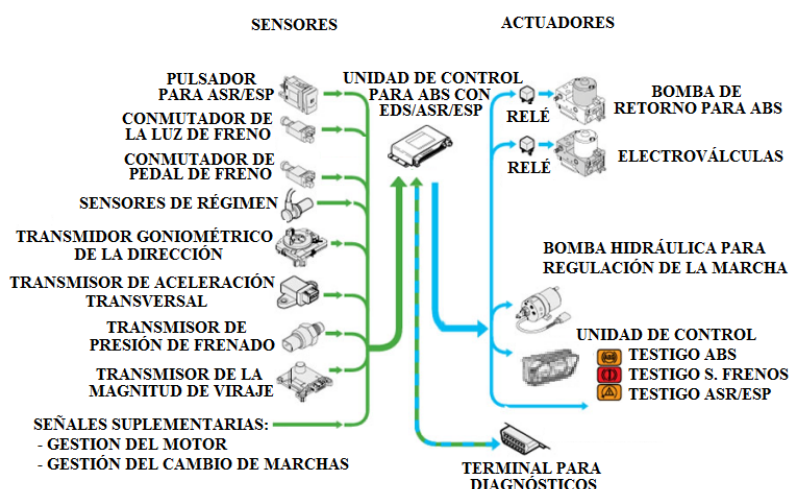


Figura 4.2-2: Mapa de flujo de información del ESP.

Cuando el vehículo está tomando una curva y los límites de adherencia con el pavimento son excedidos se produce el derrape del automóvil. Los sensores mandan a la unidad de control la información sobre los grados que está girado el volante, la velocidad a la que se desplaza el automóvil y la aceleración lateral que está experimentando. Sin embargo, la unidad conoce que bajo esas circunstancias, la aceleración transversal debería ser otra y, por tanto, el vehículo no se desplaza a lo largo de la trayectoria deseada por el conductor. Dependiendo de cuál de los dos comportamientos dinámicos siguientes esté experimentando el automóvil se actuará de una forma u otra:

- ❖ Si el vehículo tiene un comportamiento subvirador: Esto significa que el eje delantero, el que marca la dirección, pierde adherencia y, por tanto, en vez de tomar la curva tiende a salirse tangencialmente. En este caso, el ESP evita que el vehículo se salga de la curva actuando específicamente en el freno de la rueda trasera interior de la curva e interviniendo en la gestión del motor y del cambio de marchas. Figura 4.2-3.
- ❖ Si el vehículo tiene un comportamiento sobrevirador: Se produce cuando es el eje trasero es el que pierde adherencia e intenta adelantar al delantero. Este fenómeno provoca que el vehículo tienda a realizar una curva más cerrada que la determinada por la dirección del volante. En este caso, el ESP evita el derrape del vehículo actuando específicamente en el freno de la rueda delantera exterior de la curva e interviniendo en la gestión del motor y del cambio de marchas. Figura 4.2-4.



Figura 4.2-3: Situación de Subviraje



Figura 4.2-4: Situación de Sobreviraje

En casos de sobreviraje, cuando el ESP está tratando de corregir la trayectoria, muchos automovilistas realizan un exceso de contravolante que engaña al sistema y, a veces, empeora las cosas.

El sistema ESP mejora la seguridad en la conducción mediante las siguientes ventajas:

- ❖ Asistencia activa para la dirección en la conducción, incluyendo la ayuda ante condiciones críticas cuando el vehículo está sometido a fuerzas laterales importantes.
- ❖ Mejora de la estabilidad del vehículo bajo cualquier condición incluyendo frenadas repentinas, maniobras comunes de frenado o aceleración, adelantamiento y desplazamiento de carga.
- ❖ Aumento de la estabilidad del vehículo en los límites de tracción, como en maniobras extremas (como frenazos fortuitos), para reducir el peligro de derrape o choque.

- ❖ Aprovechamiento del potencial de tracción cuando el ABS y el TCS entran en acción, y cuando el MSR (controlador del par de arrastre motor) está activo, aumentando automáticamente la respuesta motora para reducir el excesivo frenado del mismo.

El resultado de estos efectos es el logro de distancias de frenado más cortas y mayor tracción, mejorando la estabilidad y consiguiendo mejores niveles de respuesta de dirección.

A las anteriores ventajas que aporta el ESP hay que sumarle otras funciones que aún siendo secundarias, facilitan, ayudan y hacen más cómoda la conducción de los vehículos hoy en día. Son las siguientes:

- ❖ **Electronic Brake Prefill (EBP):** Acerca las pastillas de los frenos a los discos en caso de que el conductor suelte el acelerador repentinamente; así, se reduce la distancia de frenada.
- ❖ **Limpieza de los Discos de Freno (BDW):** Detecta que está lloviendo a través del sensor de lluvia o si se activan los limpiaparabrisas; entonces, activa ligeramente los frenos para eliminar la humedad y suciedad de los discos de freno, reduciendo la distancia de frenada.
- ❖ **Mitigación de Vuelco del Vehículo (ROM):** Reduce el riesgo de vuelco de vehículos que tengan un centro de gravedad alto, como furgonetas o camiones de bajo peso.
- ❖ **Control adaptativo de la carga (LAC):** Ajusta las intervenciones del ESP al nivel de carga del vehículo para evitar que el peso extra pueda desestabilizarlo.
- ❖ **Mitigación del Balanceo de Remolques (TSM):** Previene el balanceo de los remolques a través de la intervención de los frenos y del par del motor en el vehículo tractor.
- ❖ **Control de la Presión de los Neumáticos (TPM-F):** Controla la presión de los neumáticos a través de sensores de velocidad de las ruedas.
- ❖ **Control de Retención en Pendientes (HHC):** Retiene el vehículo durante 1 ó 2 segundos para apoyar los arranques en cuesta después de soltar el pedal del freno.
- ❖ **Control de Descenso de Pendientes (HDC):** Función del control de velocidad para descender fuertes pendientes a baja velocidad -unos 8 km/h aprox.- y con total seguridad.

4.3 VISIÓN DE FUTURO

El automóvil no deja de evolucionar y, actualmente, muchos equipan sistemas de dirección asistida eléctrica, un motor eléctrico y una unidad de control ayudan a girar el volante, por lo que parece evidente que la siguiente generación de controles de estabilidad corrijan la trayectoria actuando sobre el volante además de contrarrestar las fuerzas de la inercia activando el freno de la rueda correspondiente. Por ejemplo, el Hyundai i40 cuenta ya con la tecnología de dirección asistida eléctrica.

Otra línea de trabajo es claramente dar respuesta a las dos preguntas que se hace el ESP - dónde quiere ir el conductor y hacia dónde va el vehículo - de una manera más exacta, rápida y al menor coste posible. Esta línea de trabajo consiste en obtener el **ángulo de deriva del vehículo, β** . Se trata de un parámetro fundamental en la dinámica del automóvil pues define el ángulo que se forma entre el eje longitudinal del vehículo y el vector velocidad, dirección hacia la que se desplaza el vehículo. Durante el presente proyecto fin de carrera se pretende explorar esta línea de investigación. En el Capítulo 5 se profundizará más acerca del parámetro beta.



5 ÁNGULO DE DERIVA DE UN VEHÍCULO



5.1 INTRODUCCIÓN

Uno de los mayores obstáculos para implementar los sistemas de control de estabilidad en los vehículos, como por ejemplo el ESP, es disponer de toda la información que pueda influir en los datos de salida del sistema, como por ejemplo el estado del vehículo (carga...), las condiciones de la carretera (adherencia, inclinación, peralte...), parámetros dinámicos del automóvil (ángulo de deriva, velocidad de guiñada, ángulo de balanceo y cabeceo...), características de los neumáticos (curva de rigidez, presión, material, dimensiones...), etc. Tener conocimiento de ellos en cada instante de tiempo es de vital importancia para mejorar los sistemas de seguridad y garantizar la estabilidad de los vehículos frente a situaciones de emergencia.

Entre todos los parámetros descritos anteriormente, el ángulo de deriva es la variable clave para conocer la dinámica del automóvil y así poder evaluar la estabilidad durante las maniobras.

Antes de desarrollar sobre el ángulo de deriva se presenta el sistema de coordenadas de un vehículo, en la Figura 5.1-1.

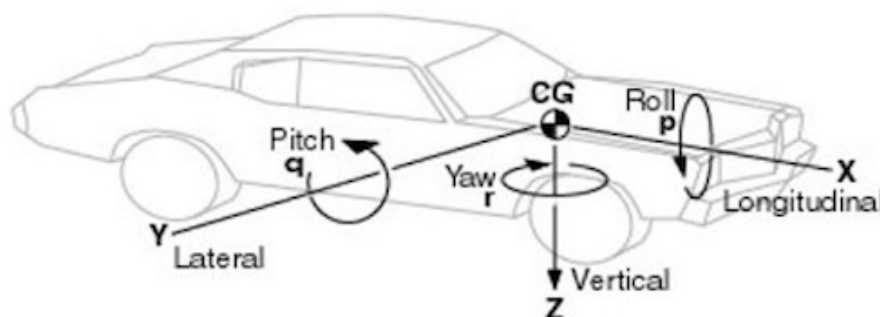


Figura 5.1-1: Sistema coordenadas vehículo.

donde:

- ❖ **Eje x:** Dirección longitudinal
- ❖ **Eje y:** Dirección transversal
- ❖ **Eje z:** Dirección vertical
- ❖ **Momento eje x:** Balanceo
- ❖ **Momento eje y:** Cabeceo
- ❖ **Momento eje z:** Guiñada

5.2 DEFINICIÓN

Cuando la trayectoria de un vehículo es en línea recta, la dirección en la que se desplaza el centro de gravedad y el eje longitudinal del vehículo (eje x) coinciden. En cambio, cuando un vehículo está tomando una curva aparece una velocidad de guiñada, provocando dos cosas: que la orientación del vehículo se vea modificada y la aparición de una aceleración lateral hacia el centro de la curva. Además, el vector velocidad deja de coincidir con la dirección longitudinal del vehículo y puede ser descompuesto sobre los ejes longitudinal (eje x), y transversal (eje y), llamada velocidad lateral (V_y), del sistema de coordenadas del vehículo.

Por el hecho de que el vector velocidad deje de coincidir con la dirección longitudinal del vehículo se crea un ángulo entre ambos. Ese ángulo es el ángulo de deriva y se denota según la letra griega “ β ” [15], [31], [36], [39], [40], [48], [49], [59], [61], [68], [70].

Por tanto, el ángulo de deriva β () es el ángulo formado entre la dirección longitudinal del vehículo, eje x , y el vector velocidad, dirección hacia la que se desplaza el automóvil. Puesto que la velocidad (vector y magnitud) depende del punto en el que se calcula, se referencia sobre el CDG del automóvil. Figura 5.2-1.

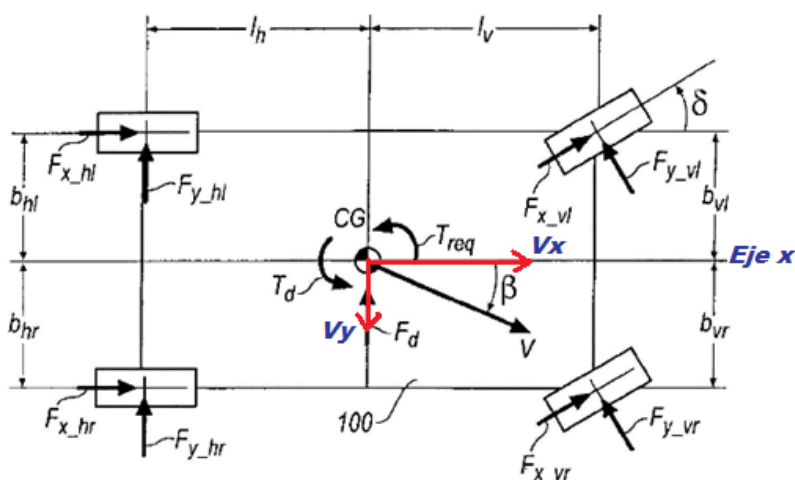


Figura 5.2-1: Sistema coordenadas vehículo.

Se tomará el ángulo beta como positivo en la dirección de las agujas del reloj y negativo en sentido antihorario, tal y como se muestra en la Figura 5.2-2.

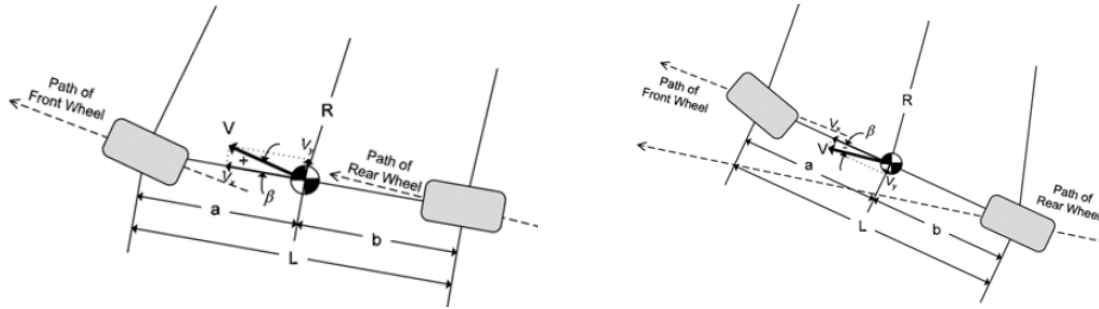


Figura 5.2-2: Esquema de signo del ángulo de deriva.

Se puede obtener el ángulo β en relación a las componentes de la velocidad según la siguiente formulación:

$$\beta = \tan^{-1} \left(\frac{V_y}{V_x} \right)$$

donde:

- ❖ β : Ángulo de deriva
- ❖ V_x : Velocidad longitudinal, sobre la componente x
- ❖ V_y : Velocidad lateral, sobre la componente y

Además, también se puede obtener en función de un punto cualquiera. Para ello, se muestra la Figura 5.2-3, la cual para mayor simplificación se representa un sistema plano del modelo del vehículo.

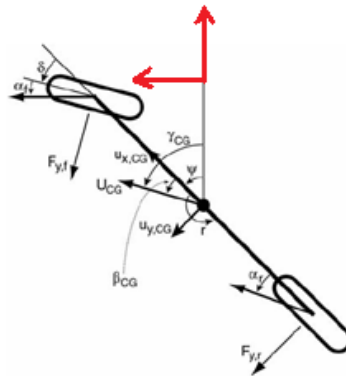


Figura 5.2-3: Sistema plano del vehículo.

$$\beta = \gamma - \psi$$

donde:

- ❖ β : Ángulo de deriva
- ❖ γ : Ángulo entre sistema de coordenadas global y vector velocidad
- ❖ Ψ : Ángulo entre el sistema de coordenadas global y eje longitudinal del vehículo.

La dinámica del vehículo está gobernada por las fuerzas de adherencia entre los neumáticos y la calzada. Por tanto, conocer las fuerzas de fricción que actúan en el vehículo es de vital importancia para conocer la dinámica del vehículo y especialmente el ángulo de deriva del vehículo. Para ello, es necesario explicar cómo se generan.

Cuando el conductor gira la dirección para realizar una curva, las ruedas directoras - habitualmente las delanteras- dejan de ser paralelas al eje longitudinal del vehículo y se desplazan lateralmente hasta que encuentran agarre con la calzada. Este proceso provoca una deformación de los neumáticos provocada por las fuerzas de adherencia, desplazando la línea media de los mismos lateralmente. Además, debido a la capacidad que tiene el neumático para recuperar su posición original se generan unas fuerzas laterales como resistencia a tal deformación permitiendo que el vehículo comience a girar. El ángulo creado entre la línea media de los neumáticos en su posición original y la nueva localización se llama ángulo de deriva de los neumáticos, análogo al ángulo de deriva del vehículo, y es denotado con la letra griega “ α ”. Ver Figura 5.2-4.

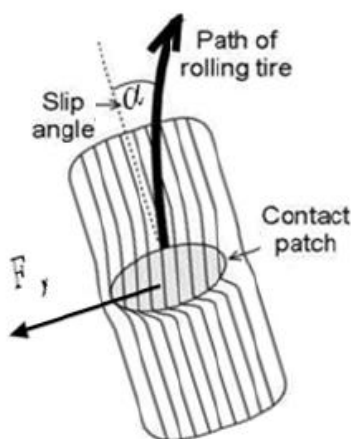


Figura 5.2-4: Ángulo de deriva de un neumático.

Durante una conducción normal, las fuerzas de laterales generadas por los neumáticos son proporcionales al ángulo de deriva de los neumáticos, siempre y cuando este no exceda de 5 grados. La expresión que define tal relación es $F_y = C_y \cdot \alpha$, donde C_y es la curva de rigidez del neumático. En cambio, en situaciones extremas, el deslizamiento de los neumáticos se incrementa notablemente y se pierde la linealidad ya que las fuerzas de adherencia se saturan y se pierde agarre entre los neumáticos y la calzada. En la Figura 5.3-2 se muestra una gráfica genérica en la que se puede observar la relación existente y explicada entre las fuerzas laterales generadas por los neumáticos y el ángulo de deriva de los mismos.

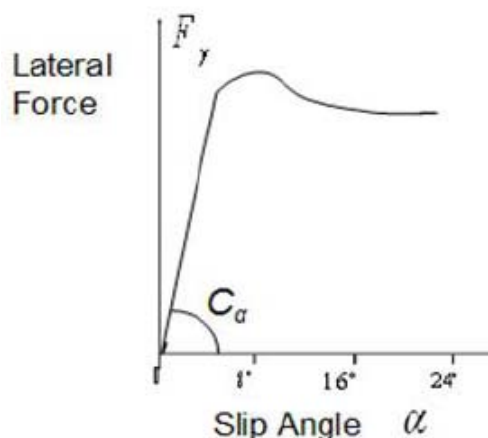


Figura 5.3-2: Relación entre el ángulo de deriva de un neumático y las fuerzas laterales.

Cuando las fuerzas de fricción en los neumáticos delanteros son saturadas antes que en las traseras, el vehículo deja de responder a la dirección marcada por el conductor y el coche se vuelve subvirador. En cambio, cuando las fuerzas de fricción en las ruedas traseras son las que primero se saturan, el coche se vuelve sobrevirador. Ver apartado 4.2.

Conocer las fuerzas laterales generadas por los neumáticos es más complicado cuando están trabajando en la zona no lineal. Existen distintos modelos basados en formulaciones empíricas derivadas de datos experimentales que relacionan todas las propiedades de los neumáticos (patrón de la banda de rodadura, material, estructura, presión, carga vertical, velocidades...) y permiten estimar el comportamiento de los neumáticos y las fuerzas laterales. Los modelos más importantes y usados son Pacejka y Dugoff.

- ❖ **Modelo Pacejka:** Este método se ha desarrollado partiendo de numerosos ensayos hasta obtener expresiones matemáticas cuyos coeficientes puedan identificarse con valores característicos de los neumáticos. La expresión matemática se conoce como “fórmula mágica” ya que predice de manera excelente el comportamiento de los neumáticos. Aún así este modelo cuenta con un gran inconveniente ya que precisa de multitud de parámetros de los neumáticos que generalmente no se conocen.
- ❖ **Modelo Dugoff:** Este modelo sintetiza todos los parámetros de los neumáticos en dos constantes: C_x y C_y , las cuales se refieren a las características longitudinales y transversales de los neumáticos respectivamente.

Relacionando el comportamiento de los neumáticos con el ángulo de deriva del vehículo se puede comentar que bajo condiciones normales de conducción, los neumáticos trabajan en la zona lineal y el vehículo responde predeciblemente a los inputs del conductor, en este caso el ángulo de

deriva del vehículo es pequeño ($\beta \leq 2^\circ$). En cambio, el ángulo de deriva alcanza valores elevados ($\beta \gg 2^\circ$) bajo condiciones límites de conducción como puede ser una maniobra evasiva de emergencia, someter al vehículo a elevadas aceleraciones/deceleraciones o cuando la adherencia entre los neumáticos y la calzada es muy bajo. En estas situaciones, los neumáticos comienzan a trabajar en la zona no lineal provocando que la dinámica del vehículo se convierta altamente no lineal y la respuesta del mismo no sea tan predecible y llegue a ser potencialmente peligrosa.

5.3 PARÁMETROS DE LOS QUE DEPENDE

Conocer en cada instante de tiempo el ángulo de deriva del vehículo es muy importante para poder mejorar e implementar nuevos sistemas de control de estabilidad del automóvil pero no es nada sencillo. A diferencia de otros parámetros que pueden ser medidos de manera sencilla, barata y directa, el ángulo beta, al depender de muchísimos parámetros, necesita sistemas bastante complejos y caros para ser obtenido lo que impide que sea implantado en los vehículos de producción. Entre otros parámetros depende de:

- ❖ **Velocidad longitudinal:** suele ser estimada a partir de la velocidad angular de las ruedas pero este valor no ofrece un valor real cuando existe una fuerte aceleración/deceleración, un elevado deslizamiento o las ruedas están bloqueadas.
- ❖ **Aceleración lateral:** es el parámetro que más influye en el ángulo de deriva del vehículo. Cuando el vehículo se desplaza a baja velocidad, la aceleración lateral es despreciable y beta es pequeño (zona lineal). En cambio, cuando se desplaza a alta velocidad, la aceleración lateral del vehículo es mayor y debe poder ser compensada por las fuerzas producidas por los neumáticos sin que lleguen a saturarse y entren en zona no lineal provocando que el ángulo de deriva aumente considerablemente.
- ❖ **Dirección:** Parámetro importante pues es la dirección marcada por el conductor y la que debe seguir el vehículo. Cuando la dinámica del vehículo se encuentra en zona no lineal, una variación en la dirección no afecta de la misma manera al momento de guiñada y es por ello que no se tiene el mismo control sobre el vehículo.
- ❖ **Velocidad de guiñada:** Velocidad angular de rotación sobre el eje vertical del vehículo, eje Z, que aparece al girar las ruedas directoras para realizar una curva.
- ❖ **Características de los neumáticos:** Curva de rigidez, presión, orientación, dimensiones...

❖ Condiciones de la carretera:

- Peralte, pendiente transversal de la calzada: influye en la velocidad lateral.
- Inclinación, pendiente longitudinal de la calzada: influye en la velocidad longitudinal.

❖ Propiedades de interfaz entre neumático y calzada: fuerzas de adherencia no lineales**❖ Características del vehículo:**

- Ángulo de balanceo: influye en la aceleración lateral.
- Ángulo de cabeceo: influye en la velocidad longitudinal.
- Carga del vehículo

A menudo, es mejor idea diseñar sistemas que estimen el ángulo de deriva (β) a partir de mediciones llevadas a cabo a bordo del vehículo, como por ejemplo: la aceleración lateral (a_y) y longitudinal (a_x), la velocidad de guiñada (ψ), la dirección (δ) y la velocidad angular de las ruedas (ω) a partir de las cuales se estima la velocidad longitudinal del vehículo (V_x), que intentar medirlo directamente debido a la gran cantidad de parámetros de los que depende y la complejidad de medirlos o conocerlos - especialmente por la no linealidad de los neumáticos y la gran influencia de las condiciones variables de la calzada.

En el capítulo 6 se detallarán las diversas metodologías encontradas en la literatura que se han desarrollado para estimar y medir el ángulo beta.



6 TÉCNICAS PARA LA MEDICIÓN Y ESTIMACIÓN DEL ÁNGULO DE DERIVA



6.1 INTRODUCCIÓN

El ángulo de deriva (β) de un vehículo es el parámetro clave para conocer el comportamiento de un automóvil, y por tanto, decisivo para mejorar los sistemas de control de estabilidad de un vehículo.

Por ese motivo, existen multitud de estudios en la literatura que tienen como objetivo obtener de la manera más rápida, precisa, exacta y al menor coste el ángulo beta. Puesto que el objetivo del presente proyecto fin de carrera es estimar el ángulo de deriva es necesario hacer un repaso sobre las diferentes técnicas usadas para la obtención del mismo.

En estudios recientes se ha conseguido medir el ángulo de deriva directamente mediante antenas GPS pero los recursos necesarios imposibilitan que sea implantado en los vehículos de producción. Es por ello que la mayoría de los estudios realizan una estimación del ángulo de deriva mediante modelos basados en la integración de sensores inerciales, relaciones cinemáticas o en redes neuronales, los cuales están en auge.

En los siguientes apartados se realizará un acercamiento a las técnicas empleadas en la literatura:

- ❖ Integración de sensores inerciales
- ❖ Relaciones cinemáticas
- ❖ Métodos combinados: Integración de sensores inerciales con Relaciones cinemáticas
- ❖ Sensores GPS
- ❖ Redes neuronales

6.2 INTEGRACIÓN DE SENSORES INERCIALES

Una técnica común para la estimación del ángulo de deriva del vehículo es mediante la integración de las medidas realizadas por sensores inerciales, los cuales proporcionan una estimación de la aceleración del vehículo. La integración de la aceleración proporciona la velocidad y ésta se puede utilizar para estimar el ángulo beta si se conoce el comportamiento del vehículo.

Este método puede proveer resultados robustos si se conocen exactamente las propiedades de los neumáticos, el coeficiente de fricción, la masa del vehículo, el momento de inercia... En cambio, puede estimar valores no reales del ángulo de deriva por varios motivos: no se conocen los parámetros anteriores, existen errores en los sensores, los parámetros medidos están deteriorados por los sensores a causa del ruido o cuando la carretera deja de ser horizontal y aparece un ángulo de inclinación o de peralte.

Este tipo de técnica ha sido considerado en las siguientes referencias: [1], [6], [7], [14], [27], [43], [50], [77], [79].

6.3 RELACIONES CINEMÁTICAS

Otra técnica, propuesta para la estimación del ángulo de deriva y quizás la más usada hasta el momento, está basada en relaciones cinemáticas. Con esta técnica se solventan los problemas aparecidos en la técnica basada en la integración de los sensores inerciales, aparte de que es más simple. No obstante, presenta los siguientes inconvenientes: solo son fiables en la región lineal y son sensibles a los cambios de los parámetros del vehículo (cambios en la carga, características de los neumáticos...), los cuales presentan una gran incertidumbre inherente y que no diferencian entre un cambio de la velocidad lateral y el efecto del ángulo de peralte de la carretera.

Diferentes tipos de modelos y estrategias son propuestas en la literatura y entre ellas se pueden encontrar las siguientes: modelos lineales, no lineales de vía única, no lineales de doble vía, etc. con observadores basados en filtros Kalman extendidos, filtros Luenberger extendidos, observadores modo deslizamiento, observadores con función de calidad adaptativa,... Los observadores Kalman y Luenberger son ampliamente utilizados ya que son idóneos para sistemas altamente no lineales como ocurre con el ángulo de deriva de un vehículo. La mayoría de los observadores diseñados están basados en un modelo de referencia pero provee buenas estimaciones solo si los parámetros son exactamente conocidos. Con idea a sobreponer esas dificultades, observadores adaptativos han sido diseñados para estimar simultáneamente el ángulo beta, las condiciones de adherencia, y la curva de rigidez del neumático.

Además, la mayoría de estos estudios están basando en modelos vehiculares lineales o cuasi-lineales. Usualmente incluyen un modelo de las fuerzas de fricción como por ejemplo en [1], [5],

[7], [8], [9], [10], [37], [43], [55], [56], [63], [75], [76], pero es importante disponer de la información sobre las condiciones de la carretera. Puesto que disponer de ello es muy complejo, algunos diseños como en [1], [5], [10], [54], [55], [80] suponen que la superficie es horizontal.

Dentro de esta técnica se han realizado multitud de estudios como por ejemplo:

En [1], [47], [63] utilizan filtros Kalman lineales o filtros Kalman extendidos para estimar beta a través de aceleración lateral y velocidad de guiñada. Los sensores ópticos proporcionan la velocidad angular de las ruedas con las que se puede obtener la velocidad lineal y los sensores inerciales proporcionan la aceleración lateral. El enfoque en [58] es similar a los anteriores pero además incluye un parámetro adaptativo, la curva de rigidez de los neumáticos, para compensar las fuerzas no lineales y obtener resultados buenos.

En [46] se presentó una típica estructura de este método basado en 4 observadores: observador lineal, un filtro Luenberger extendido, un filtro Kalman extendido, y un observador de modo de deslizamiento basado en un modelo no lineal. Dentro de las regiones lineales sus resultados eran satisfactorios pero en regiones no lineales, los resultados era erróneos debido a la hipótesis de linealidad que asumieron.

En [5] se utiliza un filtro Kalman extendido (EKF) basado en un modelo de fuerzas de fricción, el cual incluye la estimación el coeficiente de fricción y los ángulos de peralte e inclinación.

En [54] se usan alternativos EKF para estimar la velocidad y las fuerzas de los neumáticos sin el uso explícito de un modelo de fricción.

En [55] se estiman en tiempo real los parámetros de las fuerzas de fricción mediante un modelo lineal de las mismas.

En [35] no se tiene conocimiento sobre las condiciones de la carretera pero no supone que la superficie horizontal sino que incorpora el ángulo de peralte al modelo, el cual está basado en funciones de transferencia.

En [16] se usa un EKF para estimar el ángulo de deriva pero a su vez utiliza un observador lineal de entrada desconocida para estimar el ángulo de peralte que se utilizará para el modelo de fricción.

En [76] se presenta un observador no lineal que convierte los errores dinámicos observados en lineales. En [75] el observador empleado es Luenberger (LLO), una alternativa superior al EKF, especialmente cuando el sistema presenta fuertes no linealidades como es el caso.

En [61] Se compararon dos observadores: EKF y UKF. El modelo estaba diseñado en dos bloques como se muestra en la Figura 6.3-1. Del primer bloque se obtiene las fuerzas verticales y las aceleraciones en las direcciones longitudinal y transversal y en el segundo bloque se obtiene las fuerzas laterales de los neumáticos y el ángulo de deriva a partir de un modelo Dugoff de neumáticos. Las variables de entrada son tomadas a partir de sensores en la suspensión, sensores inerciales, sensores magnéticos y sensores ópticos.

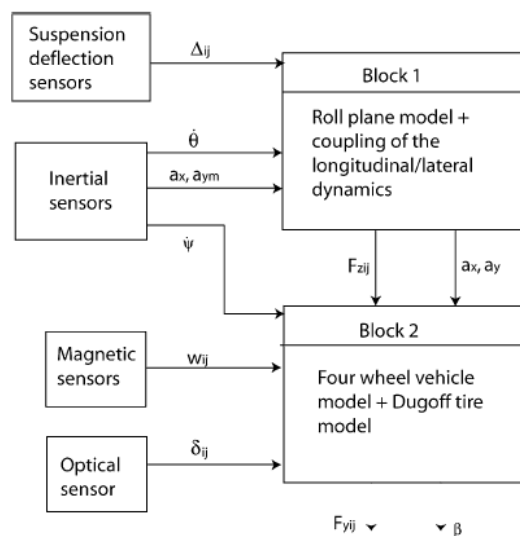


Figura 6.3-1: Esquema estudio [61].

En [56], donde se extiende el enfoque de [53] y [2], y [8] se presenta un observador que provoca que la dinámica del error no lineal estimado siga la dinámica de un sistema de referencia lineal. Asumen el conocimiento de las fuerzas de fricción longitudinales entre el neumático y la carretera.

En [9] se presenta una evolución de los estudios anteriores [56] y [8] pero obteniendo las fuerzas de fricción longitudinales a partir de la presión de frenado, posición de embrague y ángulo de acelerador.

En [40] se desarrolla un estimador de beta mediante las no linealidades de un sistema con el objetivo de reducir el tiempo computacional de un EKF al no depender de tantos parámetros además de incluir la estimación del ángulo de inclinación y peralte de la carretera. La base teórica para el diseño del observador así como algunos resultados experimentales preliminares se encuentran en [53], el cual asume el conocimiento sobre las propiedades de fricción entre la carretera y los neumáticos y que la carretera es horizontal, y en [41] para tomar en cuenta diferentes estados de la calzada. El principal problema recayó en distinguir entre un peralte distinto de cero y cuando el coeficiente de fricción es bajo.

En [51] se realiza una comparación entre un EKF y un observador, desarrollado en [52], basado en [53] y [41], al que se le ha añadido un algoritmo para la estimación del ángulo de peralte.

En [33] se realiza la estimación mediante dos bloques separados. El primero, corresponde a la dinámica del vehículo, el cual contiene un observador de modo deslizamiento (SMO). Este observador es construido con un modelo bidimensional en el que las fuerzas son modeladas sin los parámetros neumático-calzada. Esto implica una gran ventaja ya que hace robusto este bloque al obtener los valores de salida sin estar influenciados por las condiciones neumático - calzada. El segundo bloque corresponde al interfaz entre los neumáticos y la calzada. Contiene un observador Kalman extendido, desarrollado con un modelo lineal adaptativo de neumático-calzada, que estima el ángulo de deriva y la curva de rigidez de los neumáticos. En la Figura 6.3-2 se muestra un esquema de este estudio.

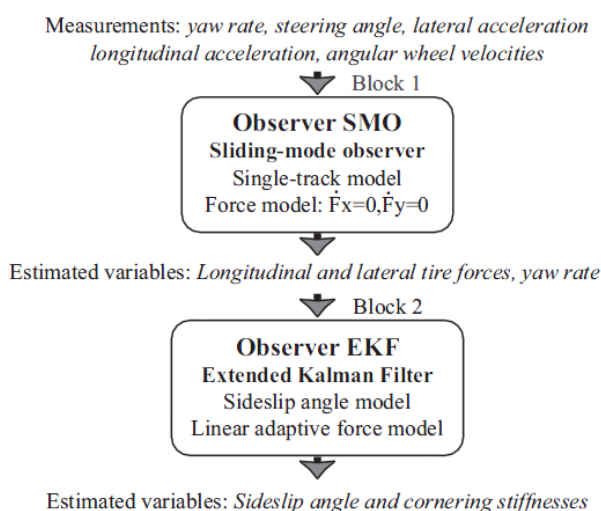


Figura 6.3-2: Esquema estudio [33].

En [48] se propuso la estimación del ángulo de deriva del vehículo mediante un estimador de bucle abierto de tres grados de libertad. Las conclusiones que obtuvieron fueron aceptable en términos generales pero cuando el vehículo se encontraba en una fuerte desaceleración, los valores obtenidos no se ajustaban debido a que los neumáticos experimentan fuerzas longitudinales adicionales produciendo torsión longitudinal en las bandas de rodadura de los neumáticos por lo que este fenómeno entraba en conflicto con la hipótesis que tomaron en la formulación del modelo ya que usaron la curva de rigidez lateral de los neumáticos para la dinámica longitudinal.

En [36] solo estimaban el ángulo de deriva cuando la velocidad de guiñada superaba el umbral de 0,5 %/s ya que pequeñas velocidades de guiñada implica pequeños ángulos de deriva.

6.4 MÉTODOS COMBINADOS

Una tercera técnica para la estimación del ángulo de deriva de un vehículo es la combinación de los dos métodos anteriores: el método basado en integración de los sensores inerciales y el método basado en relaciones cinemáticas. Son utilizados con la intención de minimizar las desventajas que cada uno de los métodos por separado tienen. En el caso de los métodos de integración pueden acumular errores en los valores de estimación debido a los sensores y al ruido de los mismos o a que no tienen en cuenta el ángulo de inclinación y el peralte de la carretera. En el caso de los métodos basados en modelos físicos son sensibles a los cambios de los parámetros del vehículo y solo son fiables en la región lineal.

En [22], el estudio propuesto está basado en esta técnica. Propone un método basado en un modelo y otro basado en relaciones cinemáticas. Las fuerzas laterales son obtenidas a partir de los valores tomados del sensor de aceleraciones laterales y del sensor de velocidad de guiñada, además de un modelo de neumáticos Dugoff acorde a un factor no lineal. Las fuerzas laterales son alimentadas en un filtro Kalman, el cual es diseñado mediante un modelo bidimensional asociado al modelo de los neumáticos. La curva de rigidez de los neumáticos compensa la no linealidad de las fuerzas de los mismos.

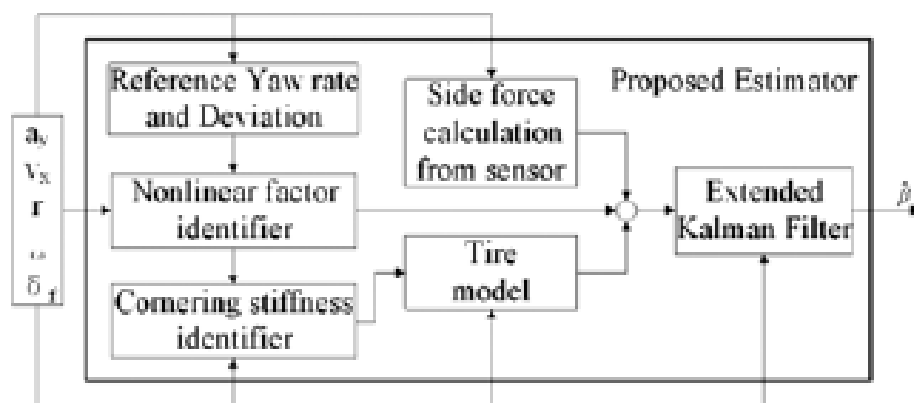


Figura 6.4-1: Esquema estudio [22].

Otros artículos publicados que usan esta técnica para la estimación del ángulo beta están recogidos en las siguientes referencias: [4], [21], [28], [80].

6.5 SENSORES GPS

No se trata de una técnica por si sola pero debido a la relevancia que tiene se ha tratado de manera independiente, como el resto de técnicas. Se trata de un método que es capaz de obtener el ángulo beta de manera directa, principal característica, sin necesidad de estimarlo. Está basado en sensores GPS que pueden proporcionar las coordenadas de posición con elevada precisión. La mayor ventaja que presentan es que pueden proporcionar ángulos de balanceo y cabeceo del vehículo cuando cuentan con más de una antena, por lo que esta técnica es excelente para realizar de manera más precisa el análisis del modelo dinámico no lineal de los vehículos. No obstante, los sensores GPS presentan varios inconvenientes: el principal de ellos es que el coste para implementar esta técnica es demasiado elevado y no permite que sea viable en vehículos de producción. Otra desventaja es que los sistemas GPS son susceptibles de perder la señal con los satélites debido a bloqueos como: túneles, grandes estructuras, el cielo cuando está cubierto... Otro inconveniente es que la velocidad de actualización es baja, lo que implica que sea inadecuado para sistemas de control avanzados en los automóviles, aún así, puede ser mejorada la tasa de refresco mediante la integración de las mediciones GPS con un sistema de navegación inercial (INS).

A continuación se muestran algunos estudios que se han llevado con sensores GPS. Algunos de ellos hacen uso de la técnica de sensores inerciales, otros basados en la técnica de relaciones cinemáticas y mediante métodos combinados.

En [17] y [18] se demostró que beta se puede estimar mediante la comparación de la integración de un giroscopio con la velocidad longitudinal de un GPS usando un filtro Kalman.

Bevly y otros autores propusieron un método mediante el sistema de posicionamiento global (GPS) y el sistema de navegación inercial (INS) basado en un modelo plano con una sola configuración. Este método no tiene en cuenta los movimientos del vehículo fuera del plano, balanceo y cabeceo, por lo que pueden conducir a estados del vehículo erróneos y realizar estimaciones de beta equivocadas.

En [49] se propone la estimación del ángulo beta mediante GPS con dos antenas y un sistema de navegación inercial, INS, los cuales están unidos a la carrocería y permiten medir los movimientos del vehículo y las condiciones de la carretera. Un filtro Kalman integra los sensores del INS con GPS para proporcionar estimaciones con elevada calidad pueden usarse para obtener beta.

En [67] se propuso una evolución de [49] donde se utilizó el sistema de posicionamiento global (GPS) y un sistema de navegación inercial (INS). Además, dos filtros Kalman, uno basado en un modelo y otro en relaciones cinemáticas, se utilizan para integrar los sensores de INS con GPS. Otros parámetros como la rigidez lateral de los neumáticos se estiman para corregir el ángulo beta basado en el modelo.

El artículo [24] propone la obtención del ángulo de deriva mediante la integración de INS con medidas obtenidas mediante GPS. Un sistemas receptor GPS con múltiples antenas pueden proveer las medidas sobre el comportamiento del vehículo con una exactitud de 0.4° . Adicionalmente, la integración del sistema de navegación inercial (INS) con GPS ofrece la ventaja de que los errores obtenidos son muy pequeños aunque su coste sea superior a los sensores inerciales.

Otros estudios que combinan medidas inerciales con medidas GPS para estimar el ángulo beta son encontrados en las siguientes referencias: [19], [20], [42], [45].

6.6 REDES NEURONALES

Por último, otra técnica que ahora está en auge es la basada en redes neuronales debido a la inherente habilidad de modelar sistemas no lineales, como es el comportamiento del ángulo de deriva de un vehículo, sin necesidad de un modelo de referencia. Estos sistemas requieren menos esfuerzos que los estimadores. El principal inconveniente que presenta esta técnicas es que no tiene en cuenta un modelo de fuerzas de fricción entre los neumáticos y la calzada.

Consiste en estimar el ángulo de deriva a partir de un patrón de variables de entrada que caracterizan la dinámica del vehículo. En una primera fase se entrena la red con patrones de entrada obtenidos mediante ensayos o mediante simulaciones vehiculares para en una segunda fase implementarla en los vehículos. De esta forma, según los valores registrados por los sensores instalados en los vehículos, el sistema será capaz de reproducir el ángulo beta asociado al patrón de variables de entrada en cada instante de tiempo.

El principal objetivo del estudio [68] era estudiar la posibilidad de aplicar redes neuronales para estimar el ángulo de deriva del vehículo asumiendo como variables de entrada la aceleración longitudinal y lateral, la velocidad de guiñada, el ángulo de dirección y la velocidad longitudinal del vehículo. Todas ellas obtenidas a bordo del vehículo a través de los sensores que suele incorporar los sistemas ESP.

En el presente proyecto se pretende avanzar en esta dirección y conocer la posibilidad de realizar la estimación del ángulo beta a partir de la técnica de redes neuronales.



7 REDES NEURONALES ARTIFICIALES



7.1 INTRODUCCIÓN

Las redes neuronales son un modelo artificial y simplificado del cerebro humano, cuya finalidad es la adquisición de conocimiento a través de la experiencia. Una red neuronal es un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona [44].

Una neurona recibe señales de todas las neuronas conectadas a ella. Estas señales se suman en el soma (núcleo de la neurona), la señal total sale codificada como pulsos eléctricos. Dichos pulsos se propagan a lo largo del axón (fibra larga y delgada) y pasan a otra neurona. Previamente, la señal es reforzada o debilitada, es decir el impulso que sale de una neurona no es igual al que entra en la siguiente neurona.

Las redes neuronales son una forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Todos los problemas que no pueden expresarse a través de un algoritmo pueden ser resueltos por medio de la experiencia.

Todos los procesos del cuerpo humano se relacionan de alguna u otra forma con la actividad o inactividad de sus neuronas.

El funcionamiento de una neurona biológica es el siguiente: la neurona es estimulada o excitada a través de sus entradas (*inputs*) y cuando se alcanza un cierto umbral, la neurona se dispara o activa, pasando una señal hacia el axón. La Figura 7.1-1 muestra el esquema de una neurona biológica.

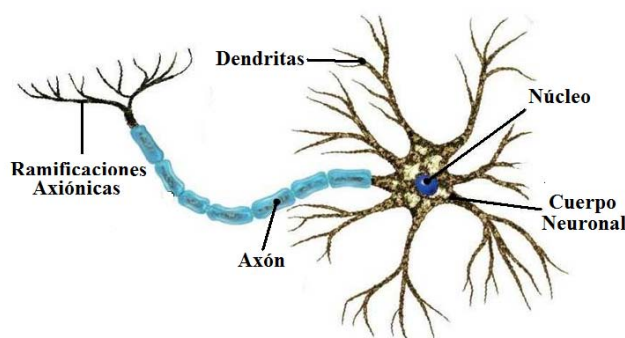


Figura 7.1-1: Neurona biológica.

Las dendritas son la vía de entrada de las señales que se combinan en el cuerpo de la neurona. De alguna manera la neurona elabora una señal de salida a partir de ellas.

El axón es el camino de salida de la señal generada por la neurona. Las sinapsis son las unidades funcionales y estructurales elementales que median entre las interacciones de las neuronas.

7.2 DEFINICIÓN

Una red neuronal artificial (RNA) es un modelo computacional inspirado en redes neuronales biológicas [29]. Especialmente por la forma de procesamiento de la información, la cual es completamente diferente al funcionamiento de un computador digital convencional. El cerebro humano corresponde al de un sistema altamente complejo, no-lineal y paralelo (capaz de realizar muchas operaciones simultáneamente) mientras que los ordenadores actuales son de tipo secuencial (solo realizan una operación cada vez). Las RNA se caracterizan por:

- ❖ Aprendizaje a través de la experiencia, el cual es almacenador en el peso relativo de las conexiones interneurales.
- ❖ Altísima plasticidad y adaptabilidad lo que significa que son capaces de cambiar dinámicamente junto con el medio.
- ❖ Tener un comportamiento altamente no lineal, lo que les permite procesar información procedente de otros fenómenos no-lineales.
- ❖ Alto nivel de tolerancia a fallas, es decir, pueden sufrir un daño considerable y continuar teniendo un buen comportamiento.

Debido a estas características y muchas otras, las neuroredes se han convertido en una gran ayuda en el procesamiento de datos experimentales de comportamiento complejo. Además, su comportamiento iterativo no-lineal las une de modo natural al caos de teorías de la complejidad. De hecho, las posibilidades son tan amplias que se empieza a hablar de un nuevo campo, aparte de la Biología, la Matemática y la Física: las Neurociencias [30].

7.3 ELEMENTOS BÁSICOS

La estructura de una red neuronal está caracterizada por las conexiones de unas unidades con otras que se disponen en forma de capas. En la Figura 7.3-1 se muestran las diferentes capas que componen una RNA:

- ❖ **Capa de entrada:** Las neuronas que pertenecen a este grupo, no reciben como entrada la salida de ninguna otra neurona y, su salida es la entrada a otras neuronas de la red. Por tanto, esta capa actúa como entrada al sistema.
 - ❖ **Capa de salida:** Las neuronas situadas en esta capa, reciben como entrada la salida de otras neuronas pero no es la entrada de ninguna otra capa. Por lo tanto, se tratan de salidas de la información para que puedan estas ser leídas.
 - ❖ **Capa oculta o intermedia:** una red neuronal puede estar formada por varias capas ocultas. Reciben como señal la salida de otras neuronas y, a su vez, su salida es la entrada de
-

otra capa sucesiva. Son las encargadas de extraer, procesar y memorizar la información concerniente al problema estudiado, basándose para ello principalmente en los pesos de las conexiones de la red.

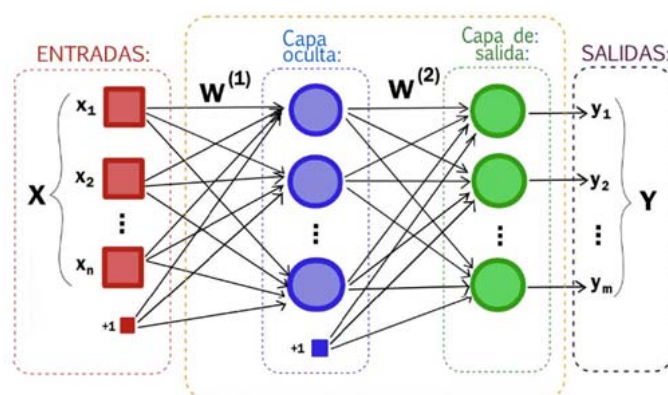


Figura 7.3-1: Esquema de una RNA.

7.4 FUNCIONES

7.4.1 FUNCIÓN DE ENTRADA

La función de entrada se conoce como función de propagación o ponderación, y se encarga de transformar las diferentes entradas que provienen de la sinapsis en el potencial de la neurona. Algunas de las funciones de entrada más utilizadas y conocidas son:

- ❖ **Suma:** Suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos:

$$\sum_j (n_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n \quad \text{Ec. 2.5.4.1-1}$$

- ❖ **Producto:** Se trata del producto de los valores obtenidos al multiplicar cada entrada a la neurona por su peso correspondiente:

$$\prod_j (n_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n \quad \text{Ec. 2.5.4.1-2}$$

- ❖ **Máximo de las entradas pesadas:** solamente toma en consideración el valor de entrada más fuerte, previamente multiplicado por su peso correspondiente:

$$\text{Max}_j (n_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n \quad \text{Ec. 2.5.4.1-3}$$

7.4.2 FUNCIÓN DE ACTIVACIÓN

La función de activación es la característica principal de una neurona. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total a la misma.

Las funciones más comunes son las siguientes: [26]

a) **Lineal:** La activación de la neurona es proporcional a la entrada total. La función lineal más típica es la que se muestra en la Figura 7.4.2-1.

Este tipo de función produce comportamientos neuronales inestables debido a que sus activaciones tienden a incrementarse sin límites. Lo ideal es que la función de activación tenga umbral para que cuando el estímulo total exceda de un cierto valor la salida sea constante.

b) **Escalón:** La activación es un valor discreto, (0,1) que depende de si la entrada total supera o no un determinado umbral. Presenta el inconveniente de no ser derivable en cero, por lo tanto, se limita su capacidad de aprendizaje y representación. La funciones signo y escalón forman parte de este tipo. Figura 7.4.2-1.

c) **No lineales:** En las funciones no lineales, la salida no es proporcional a la entrada. Las redes con salida continua usan esta función ya que su algoritmo de aprendizaje necesita una función derivable. En este grupo destacan la función logística, la función tangente hiperbólica y la sinusoidal. Figura 7.4.2-1.

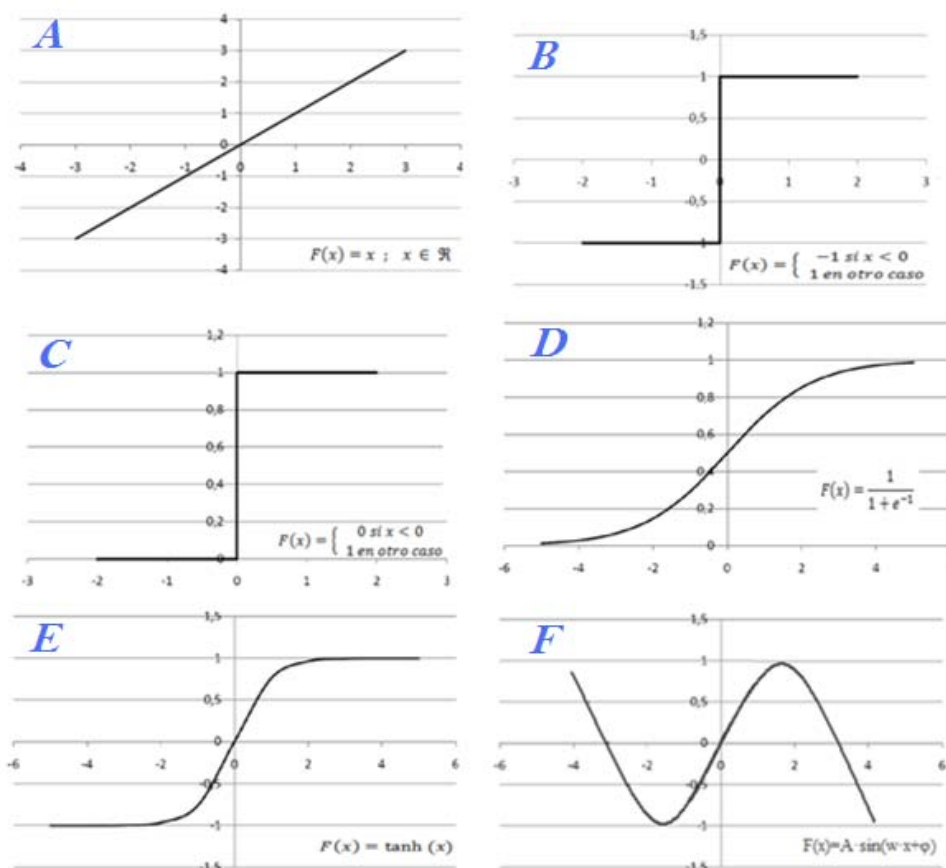


Figura 7.4.2-1: Funciones de activación. A) Función lineal. B) Función signo. C) Función escalón. D) Función no lineal hiperbólica. E) Función no lineal tangente. F) Función no lineal sinusoidal.

7.4.3 FUNCIÓN DE SALIDA

La función de salida convierte el estado de la neurona en la salida hacia la siguiente neurona que se transmite por la sinapsis, Figura 7.4.3-1. Generalmente, no se considera y se toma la identidad, de manera que la salida es el propio estado de activación de la neurona, $Z_j=Y_j$.

Existen algunas redes que transforman su estado de activación en una salida binaria y para eso usan la función escalón como salida.

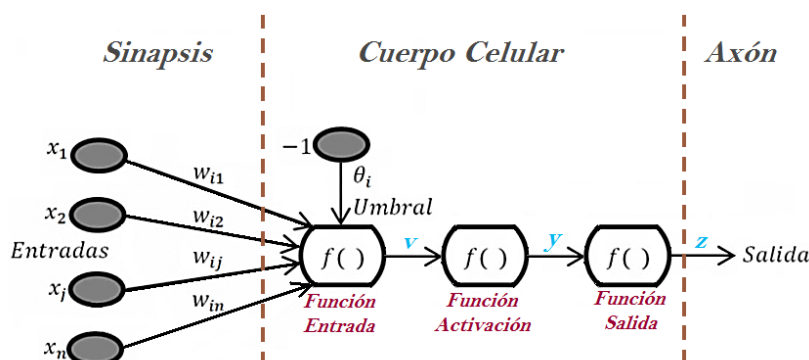


Figura 7.4.3-1: Funcionamiento de una neurona de la red.

7.5 MODO DE OPERACIÓN

El modo de operación se refiere a la manera en que la red neuronal procesa los estímulos externos y crea la respuesta de salida. Puede ser una red:

- ❖ **Red estática:** No tiene en cuenta el valor de las salidas en el instante anterior por lo que para unos valores de entrada establecidos siempre se obtienen las mismas salidas. Estas redes no cuentan con elementos de realimentación y/o retardo. Debido a su modo de funcionamiento, estas redes tienen una capacidad limitada para sintetizar funciones dependientes del tiempo en comparación con las redes binarias.
- ❖ **Red dinámica:** Sí tiene en cuenta el valor de las salidas en instantes de tiempo anteriores por lo que para unos valores de entrada establecidos no necesariamente se obtienen las mismas salidas. Son idóneas en aquellos casos en los que la variable tiempo está presente pero, en cambio, plantean problemas de convergencia y estabilidad difíciles de analizar. Las realimentaciones pueden ser de dos maneras:
 - **De la salida:** Las salidas son realimentadas a las capas anteriores.
 - **Del estado:** Las salidas de las capas ocultas son realimentadas a las capas anteriores.

7.6 OPERACIONES DE CAPA

Dependiendo de cómo trabaje la capa en su conjunto se pueden diferenciar dos tipos de operaciones:

- ❖ **Normalización:** Considera la salida de cada elemento de la capa ajustándolas para dar un nivel constante de actividad.
- ❖ **Competencia:** No todas las salidas de cada elemento de la capa contribuyen para la generación de la salida.

7.7 OTRAS CONSIDERACIONES

También es posible caracterizar las redes neuronales teniendo en cuenta la forma de operar a la hora de generar la salida o al actualizar los pesos. Así se tiene:

- ❖ **Operación síncrona:** cuando todos los elementos de proceso del sistema generan la salida a la vez.
- ❖ **Operación asíncrona:** cuando los elementos de proceso que la constituyen generan la salida aleatoriamente e independientemente unos de otros. En estos casos puede añadirse a los elementos de proceso entradas de control que indiquen cuando han de ser actualizados los pesos de sus conexiones con los otros elementos de proceso. Hay que tener en cuenta que, en muchos modelos de redes, la actualización de los pesos requiere la ejecución de la red a fin de calcular el correspondiente error, por lo que el cambio en el valor de los pesos también queda influido por el modo de operación síncrona o asíncrona.

7.8 CÁLCULO DEL ERROR

Para saber la precisión de los resultados proporcionados por la red neuronal, se recurre a una función que calcula el error existente entre la salida estimada por la red y el valor de la salida que se le proporcionó. Como la salida de la red depende de forma directa del valor de los pesos en un determinado instante, se puede afirmar que el error cometido es función de los pesos de la red. El algoritmo de aprendizaje tratará de buscar un conjunto de pesos que minimice dicho error.

Algunas medidas estándar del error son:

❖ **La suma de los cuadrados de los errores (Sum of Square Errors, SSE):**

$$SSE = \sum_p \sum_i (Y_{Pi} - \hat{Y}_{Pi})^2 \quad \text{Ec. 7.8-1}$$

donde:

- p: pertenece al conjunto de patrones.
- i: número de neuronas en la capa de salida.
- \hat{Y}_p : salida proporcionada por la red ante la presentación del patrón de entrada Y_p .

El error SSE de entrenamiento indica el error final alcanzado por todos los patrones que son utilizados durante el proceso de entrenamiento de red, mientras que el error SSE de validación define el error obtenido utilizando los patrones de validación sin propagar dichos valores en la red, es decir, manteniendo los pesos en las neuronas que fueron alcanzados con anterioridad durante el periodo de aprendizaje.

❖ **El error cuadrático medio (Mean Square Error, MSE):**

$$MSE = \frac{\sum_p \sum_i (Y_{Pi} - \hat{Y}_{Pi})^2}{r} = \frac{SSE}{r} \quad \text{Ec. 7.8-2}$$

donde: r es el número de patrones utilizados en el aprendizaje.

❖ **La raíz cuadrada del error cuadrático medio (Root Mean Square Error, RMSE):**

$$RMSE = \sqrt{\frac{\sum_p \sum_i (Y_{Pi} - \hat{Y}_{Pi})^2}{r}} = \sqrt{MSE} \quad \text{Ec. 7.8-3}$$

7.9 MECANISMOS DE APRENDIZAJE

Para que una red neuronal sea apta debe ser capaz de obtener los valores de salida correctos para cualquier conjunto de datos de entrada. Para ello, en una primera etapa, proceso de entrenamiento, se introducen un conjunto de valores de entrada, datos de entrenamiento. La red neuronal los memoriza y obtiene generalizaciones con los que obtener los valores de salida, sean cuales sean las entradas.

Una red habrá aprendido cuando las funciones de cada neurona (entrada, activación y salida) se mantengan constantes y solo los pesos sobre cada una de las conexiones varíen. Es por ello que la clave del aprendizaje de una red neuronal está en la adaptación de los pesos.

$$\frac{\partial w_{ij}}{\partial t} = 0 \quad \text{Ec. 7.9-1}$$

En los sistemas biológicos, existe una continua destrucción y creación de conexiones entre las neuronas. En los modelos de redes neuronales artificiales se traduce al valor que adoptan los pesos de cada conexión. Un peso con valor distinto de cero supone la creación de una nueva conexión, mientras que un peso de valor cero rompe la conexión.

Para el aprendizaje de las redes neuronales es importante conocer cuáles son los criterios que se siguen para cambiar el valor asignado a los pesos (las conexiones). Existen tres métodos de aprendizaje importantes [3]:

- ❖ Aprendizaje supervisado.
- ❖ Aprendizaje no supervisado.
- ❖ Aprendizaje por refuerzo.

7.9.1 APRENDIZAJE SUPERVISADO

El aprendizaje supervisado se caracteriza porque se realiza mediante un entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada. Se controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime lo máximo posible a la deseada.

Dentro del aprendizaje supervisado podemos encontrar:

- ❖ **Aprendizaje por corrección del error:** consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red (en función del error cometido en la salida). Se pueden destacar los siguientes algoritmos:
 - **Regla de aprendizaje Perceptron:** para cada neurona en la capa de salida se calcula la desviación a la salida, conocida como el error (δ), con el objetivo de utilizarlo para modificar los pesos en la conexión de la neurona precedente.

- **Regla de aprendizaje Delta o Least Mean Squared Error (LMS-Error):** También llamado mínimo error cuadrático. Utiliza la desviación a la salida pero toma en consideración a todas las neuronas predecesoras que tiene la neurona de salida. Esto permite cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red ya que cuanto más información se tenga sobre el error cometido, más rápido se puede aprender.
- **Regla de aprendizaje de propagación hacia atrás o de Backpropagation:** También conocido como regla LMS multicapa. Se trata de una generalización de la regla de aprendizaje Delta. Se desarrollará en el Capítulo 8 ya que será empleada en este proyecto.
- ❖ **El aprendizaje por refuerzo:** Es más lento que el anterior. Consiste en no proporcionar el valor de salida que debería obtenerse ante una determinada entrada durante el entrenamiento. En este tipo de aprendizaje la función del supervisor es indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 ó fracaso = -1)
- ❖ **El aprendizaje estocástico:** consiste en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

7.9.2 APRENDIZAJE NO SUPERVISADO

Las redes con aprendizaje no supervisado o autosupervisado, no precisan influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada.

Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. Se suelen considerar dos tipos de algoritmos de aprendizaje no supervisado:

- ❖ **Aprendizaje hebbiano:** Se basa en que si dos neuronas N_i y N_j toman el mismo estado simultáneamente (ambas activas o inactivas), el peso de la conexión entre ambas se incrementa. Las neuronas son binarias por lo que las entradas y salidas permitidas a la neurona son: $\{-1, 1\}$ ó $\{0, 1\}$. Esto se debe a que la regla de aprendizaje de Hebb se originó a partir de la neurona biológica clásica, que solamente puede estar activa o inactiva.

- ❖ **Aprendizaje competitivo y comparativo:** Si se detecta que un patrón nuevo que pertenece a una clase reconocida previamente, entonces la inclusión de este nuevo patrón a esta clase matizará la representación de la misma. Por el contrario, si no pertenece a ninguna de las clases reconocidas anteriormente, entonces la estructura y los pesos de la red neuronal serán ajustados para reconocer la nueva clase.

7.9.3 APRENDIZAJE ON-LINE Y OFF-LINE

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual (on-line) o si el aprendizaje supone la inhabilitación de la red hasta que el proceso termine (off-line). Cuando el aprendizaje es *off-line*, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de test o prueba, que serán utilizados en la correspondiente fase. Una vez terminada la etapa de entrenamiento de la red los pesos de las conexiones permanecen fijos. Debido a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento.

8 APRENDIZAJE BACKPROPAGATION



8.1 INTRODUCCIÓN

El algoritmo de aprendizaje de *Backpropagation* es un tipo de regla de aprendizaje supervisado, que emplea ciclos de propagación. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga a través de las capas intermedias de la red hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula el error para cada una de ellas [29] [64].

El error se propaga hacia atrás, desde la capa de salida hacia todas las neuronas de la capa oculta. Sin embargo, cada neurona de las capas ocultas sólo recibe una fracción de la señal total del error dependiendo de la contribución relativa que haya aportado a la salida original. Posteriormente, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento y, así, aprendan a reconocer las características de entrada. Con el proceso anterior se consigue que cuando un patrón de entrada incompleto se asemeja a uno reconocido por la red, las neuronas de la capa oculta responderán para obtener una salida activa. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para la que han sido entrenadas.

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la red *Backpropagation* tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento en clases. Esta tendencia se puede extrapolar, para llegar a la hipótesis de que todas las unidades de la capa oculta de una *Backpropagation* son asociadas de alguna manera a características específicas del patrón de entrada como consecuencia del entrenamiento. La asociación puede no resultar evidente para el observador humano, lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le ofrece las entradas, en el proceso de entrenamiento. Esta misma representación interna se puede aplicar a entradas que la red no haya visto antes ya que las clasificará según las características que comparten con los ejemplos de entrenamiento.

8.2 REGLA DE APRENDIZAJE

Para que la red *Backpropagation* trabaje es necesario introducir los datos de entrenamiento de la siguiente forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_q, t_q\} \quad \text{Ec. 8.2-1}$$

Donde p_q es una entrada a la red y t_q es la correspondiente salida deseada para el patrón q -ésimo, necesaria puesto que se trata de un aprendizaje supervisado como ya se definió en el

apartado 7.9.1. El algoritmo debe ajustar los parámetros de la red para minimizar el error medio cuadrático.

Además de introducir los datos es necesario definir la tipología de la red: número de capas, de neuronas en cada una de ellas, las conexiones y las funciones de transferencia. Hasta ahora no existe una técnica para determinar cómo debe ser la tipología por lo que será determinada por la experiencia del diseñador. Este deberá tener en consideración que cuanto más compleja sea, más tiempo tardará la red en obtener la salida. Por ello, es preciso llegar a un compromiso.

Una vez calculada la salida, se compara con el valor objetivo y se calcula el error. Este error marca el camino más adecuado para la actualización de los pesos y ganancias que al final del entrenamiento producirán una respuesta satisfactoria a todos los patrones de entrenamiento. Esto se logra minimizando el error medio cuadrático en cada iteración del proceso de aprendizaje.

Cuando se presenta a la red un patrón de entrenamiento, este se propaga a través de las conexiones existentes produciendo una entrada n en cada una de las neuronas de la siguiente capa. La entrada a la neurona j de la siguiente capa se debe a la presencia de un patrón de entrenamiento en la entrada:

$$n_j^0 = \sum_{i=1}^q W_{ji}^0 + b_j^0 \quad \text{Ec. 8.2-2}$$

donde:

- ❖ W_{ij}^0 es el peso que une la componente i de la entrada con la neurona j de la primera capa oculta.
- ❖ p_i es la componente i del vector p que contiene el patrón de entrenamiento de q componentes.
- ❖ b_j^0 es la ganancia de la neurona j de la capa oculta.

El superíndice $(^0)$ representa la capa a la que pertenece cada parámetro, en este caso a la capa oculta.

Cada una de las neuronas de la capa oculta tiene como salida a_j^0 , definida según la siguiente ecuación:

$$a_j^0 = f^0 \sum_{i=1}^q W_{ji}^0 p_i + b_j^0 \quad \text{Ec. 8.2-3}$$

donde: f^0 es la función de transferencia de las neuronas de la capa oculta.

Las salidas a_j^0 de las neuronas de la capa oculta son las entradas a los pesos de conexión de la capa de salida, $a_k^0 \longrightarrow n_k^s$, este comportamiento está descrito por la ecuación:

$$n_k^s = \sum_{i=1}^m W_{ki}^s a_i^0 + b_k^s \quad \text{Ec. 8.2-4}$$

donde:

- ❖ W_{kj}^s es el peso que une la neurona j de la capa oculta con la neurona k de la capa de salida, la cual cuenta con s neuronas.
- ❖ a_j^0 es la salida de la neurona j de la capa oculta, la cual cuenta con m neuronas.
- ❖ b_k^s es la ganancia de la neurona k de la capa de salida.
- ❖ n_k^s es la entrada a la neurona k de la capa de salida.

La red produce una salida final descrita por la ecuación:

$$a_k^s = f^s(n_k^s) \quad \text{Ec. 8.2-5}$$

donde f^s es la función de transferencia de las neuronas de la capa de salida.

A partir de las ecuaciones Ec. 8.2-4 y Ec. 8.2-5 se obtiene la salida de la red en función de la entrada y de los pesos de conexión con la última capa oculta:

$$a_k^s = f^s \left(\sum_{j=1}^m W_{ki}^s a_j^0 + b_k^s \right) \quad \text{Ec. 8.2-6}$$

La salida de la red de cada neurona se compara con la salida deseada para calcular el error en cada unidad de salida:

$$\delta_k = (t_k - a_k^s) \quad \text{Ec. 8.2-7}$$

El error debido a cada patrón p propagado está dado por:

$$ep^2 = \frac{1}{2} \sum_{k=1}^s (\delta_k)^2 \quad \text{Ec. 8.2-8}$$

donde: δ_k es el error en la neurona k de la capa de salida.

Este proceso se repite para el número total de patrones de entrenamiento (r). Para un proceso de aprendizaje, el objetivo del algoritmo es actualizar todos los pesos y ganancias de la red minimizando el error medio cuadrático total:

$$e^2 = \sum_{p=1}^r ep^2 \quad \text{Ec. 8.2-9}$$

donde: e^2 es el error total en el proceso de aprendizaje en una iteración, después de haber presentado a la red los r patrones de entrenamiento.

El error que genera una red neuronal en función de sus pesos, genera un espacio de n dimensiones, donde n es el número de pesos de conexión de la red. Al evaluar el gradiente del error en un punto de esta superficie se obtendrá la dirección en la cual la función del error tendrá un mayor crecimiento.

Como el objetivo del proceso de aprendizaje es minimizar el error, debe tomarse la dirección negativa del gradiente para obtener el mayor decremento del error y de esta forma su minimización, condición requerida para realizar la actualización de la matriz de pesos en el algoritmo *Backpropagation*:

$$W_{k+1} = W_k - \alpha \nabla e p^2 \quad \text{Ec. 8.2-10}$$

Después de encontrar el valor del gradiente del error, se procede a actualizar los pesos de todas las capas empezando por la de salida. Para la capa de salida la actualización de pesos y ganancias está dada por:

$$W_{kj}(t+1) = W_{kj}(t) - 2\alpha \delta_k^s \quad \text{Ec. 8.2-11}$$

$$b_k(t+1) = b_k(t) - 2\alpha \delta_k^s \quad \text{Ec. 8.2-12}$$

donde: α es el parámetro de aprendizaje que define la velocidad de aprendizaje.

Después de actualizar los pesos y ganancias de la capa de salida se procede a actualizar los pesos y ganancias de la capa oculta mediante las ecuaciones:

$$W_{ki}(t+1) = W_{ki}(t) - 2\alpha \delta_k^0 p_i \quad \text{Ec. 8.2-13}$$

$$b_j(t+1) = b_j(t) - 2\alpha \delta_k^0 \quad \text{Ec. 8.2-14}$$

Esta deducción fue realizada para una red de tres capas.

En las técnicas de gradiente descendiente es conveniente avanzar por la superficie de error mediante incrementos de los pesos. Si se toman incrementos grandes, debido a que no se conoce lo lejos o lo cerca que se está del punto mínimo, se corre el riesgo de pasar por encima del punto mínimo y estar oscilando alrededor de él sin llegar a alcanzarlo. En cambio, si se coge un incremento pequeño, se encontrará la solución pero se verá penalizada la velocidad de convergencia del algoritmo. Por tanto, para un mejor funcionamiento y una mayor velocidad de convergencia, se

define un incremento variable según el parámetro de aprendizaje α . Se comenzará con valores elevados y se irá disminuyendo a medida que se vaya acercando al punto mínimo buscado, garantizando así una rápida convergencia.

En el desarrollo matemático que se ha realizado para llegar al algoritmo *Backpropagation*, no se asegura en ningún momento que el mínimo que se encuentre sea global. En todo caso, si la solución es admisible desde el punto de vista del error, no importa si el mínimo es local o global o si se ha detenido en algún momento previo a alcanzar un verdadero mínimo.

A continuación se expondrá de forma gráfica la deducción matemática alcanzada en el punto anterior. Para ello se utilizará una red formada por dos entradas, una salida y dos capas ocultas de neuronas. La estructura de red utilizada se observa en la Figura 8.2-1, [29].

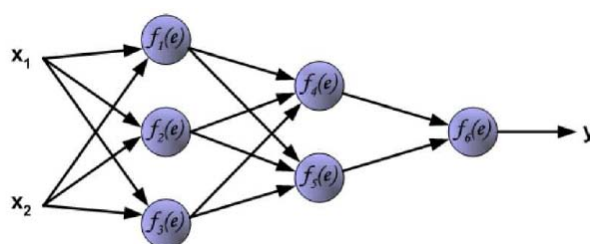


Figura 8.2-1: Estructura de una red.

Cada neurona está compuesta por dos unidades; la primera unidad representa el coeficiente de pesos de las neuronas de entrada, mientras que la segunda unidad representa la función de activación de la neurona, Figura 8.2-2, donde e es la suma de las señales de entrada, mientras que $y=f(e)$ es la función de salida.

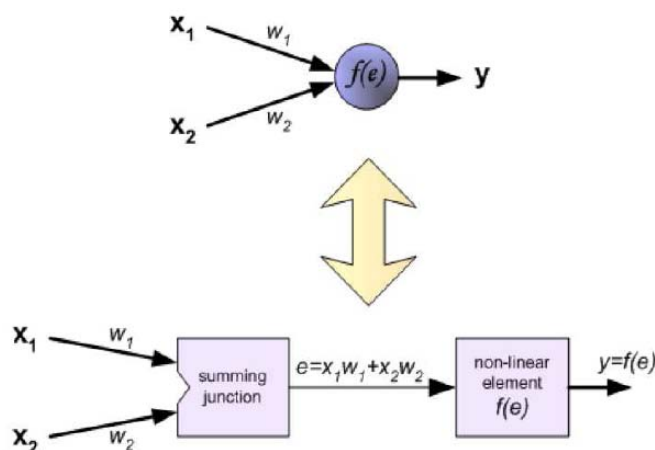


Figura 8.2-2: Componentes de una neurona.

Se trata de un proceso iterativo y en cada iteración los coeficientes de los pesos son modificados usando los datos de entrenamiento. La modificación se calcula usando el procedimiento descrito a continuación: cada ciclo comienza con las señales de entrada en una neurona entrada (x_1 y x_2), formándose la señal de salida de dicha neurona, y así se puede determinar la salida para cada capa.

La Figura 8.2-3 muestra la propagación de la señal a través de la red, donde $W_{(xm)n}$ representa los pesos de cada neurona x_m de la capa n , e y_n representa la señal de salida de la neurona n .

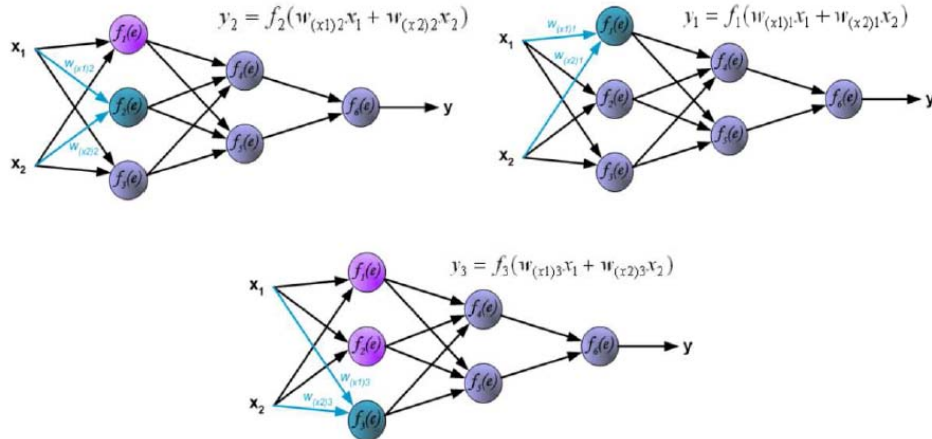


Figura 8.2-3: Reparto de pesos en la primera capa de neuronas.

La señal de propagación atraviesa la capa oculta como se puede observar en la Figura 8.2-4, donde w_{mn} representa los pesos de conexiones entre la salida de la neurona m y la entrada de la neurona n de la siguiente capa. La señal continúa propagándose hasta alcanzar la capa de salida, Figura 8.2-5.

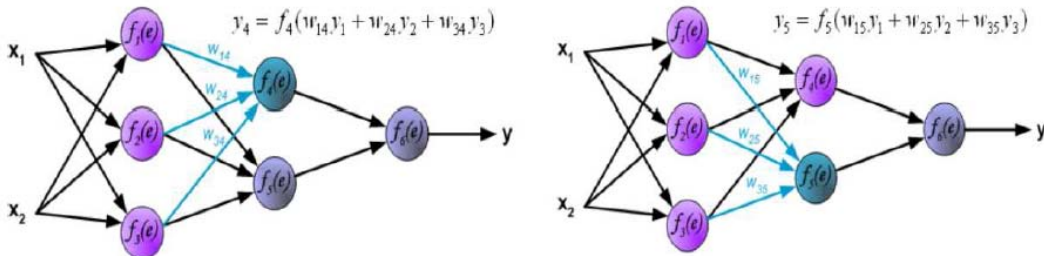


Figura 8.2-4: Repartos de pesos en la segunda capa de neuronas.

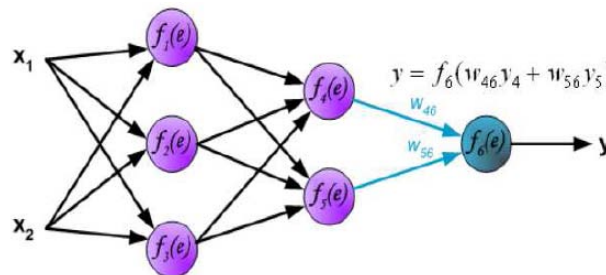


Figura 8.2-5: Señal de salida.

El siguiente paso del algoritmo consiste en comparar el valor de la señal de salida con el valor deseado. La diferencia entre ambas es conocida como la señal de error δ de la capa de salida, Figura 8.2-6. El error se propaga en dirección contraria, es decir, atravesado todas las capas intermedias hasta alcanzar las neuronas de la capa de entrada, Figuras 8.2-7 y 8.2-8.

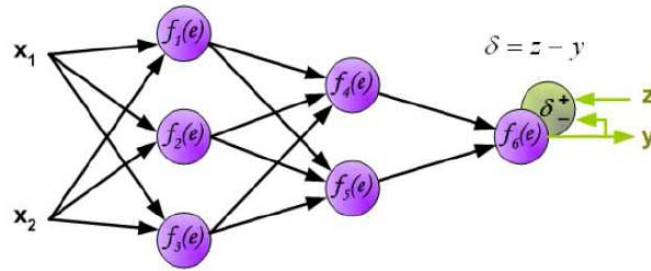


Figura 8.2-6: Diferencia entre la salida obtenida y la salida deseada.

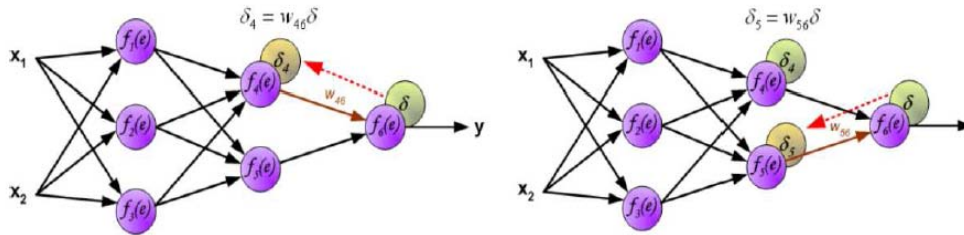


Figura 8.2-7: Propagación del error en la capa intermedia.

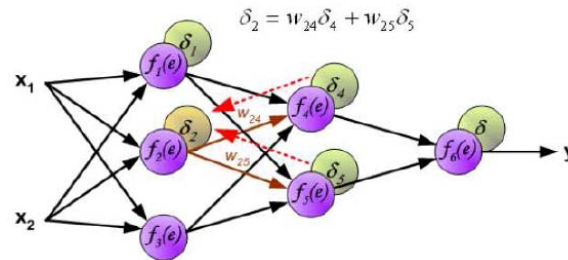


Figura 8.2-8: Propagación del error en la primera capa de la red.

Cuando la señal de error ha sido propagada por toda la red, los pesos son modificados según las Figuras 8.2-9 hasta Figura 8.2.11, donde n es el factor de aprendizaje. Este proceso es iterativo hasta que la señal de salida obtenida sea la deseada.

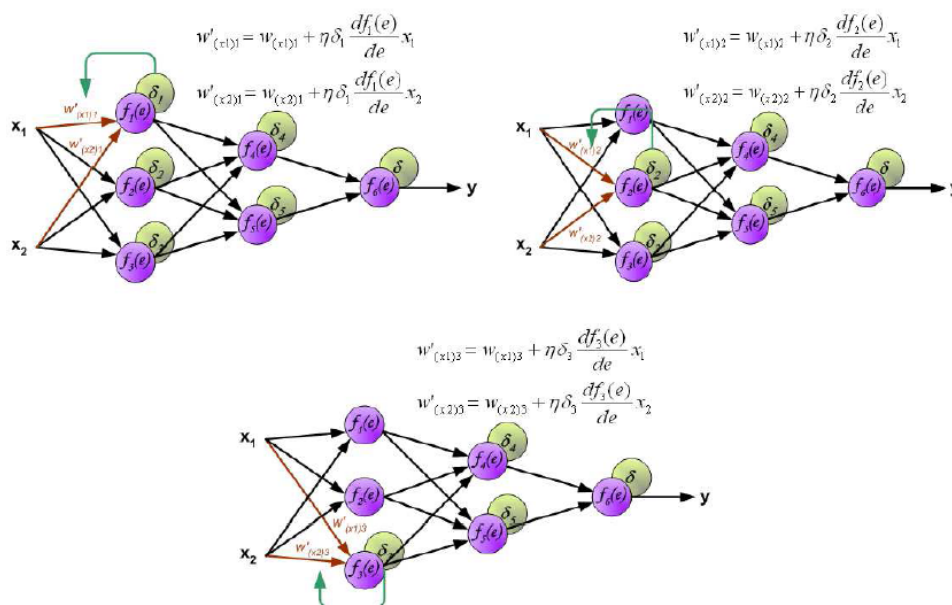


Figura 2.6.2-9: Modificación de los pesos en la capa de entrada.

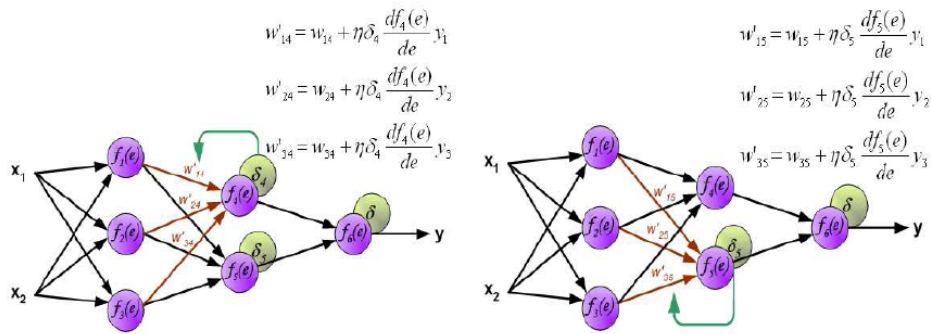


Figura 2.6.2-103: Modificación de los pesos en la capa intermedia.

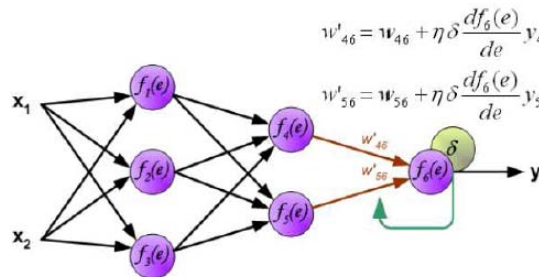


Figura 2.6.2-11: Propagación del error hasta la capa de salida.

9 SOFTWARE EMPLEADO



9.1 INTRODUCCIÓN

Para la realización del presente proyecto fin de carrera se ha hecho uso de dos software:

- ❖ **CarSim:** Empleado para simular las diferentes maniobras con un vehículo y extraer los parámetros necesarios con los que estimar el ángulo de deriva.
- ❖ **Java-NNS:** Empleado para realizar los análisis mediante redes neuronales. Los datos de entrada serán los proporcionados por el programa CarSim. [72]

9.2 CARSIM

CarSim es un paquete de software comercial que permite predecir el rendimiento de los vehículos, en respuesta a los controles del conductor (dirección, acelerador, frenos, embrague y cambio de marchas) en un entorno determinado (geometría de la vía, los coeficientes de fricción, viento...). CarSim es producida y distribuida por una compañía norteamericana, Mechanical Simulation Corporation, utilizando la tecnología que se originó en la Universidad de Michigan Transportation Research Institute (UMTRI) en Ann Arbor, Michigan.

La versión de CarSim que se ha utilizado es la 8.02.

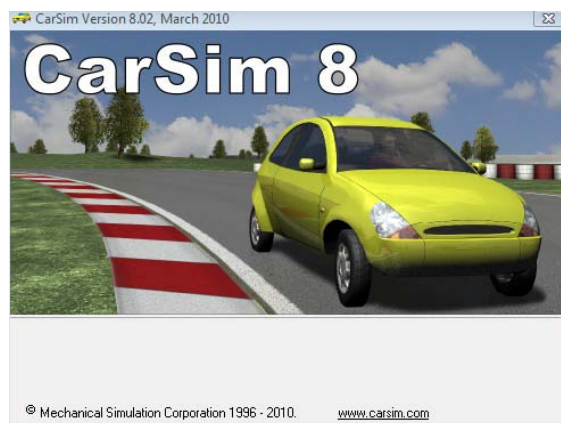


Figura 9.2-1: CarSim.

CarSim simula el comportamiento dinámico de los vehículos (de pasajeros, carreras, camiones ligeros, vehículos utilitario...) además de permitir el diseño, desarrollo, prueba y planificación de programas de vehículos. Para ello, cuenta con más de ochocientas variables que posteriormente pueden ser representadas en gráficos o exportadas a otros programas como MATLAB, Excel y herramientas de optimización para su análisis.

Los modelos matemáticos con los que se realizan los cálculos internos han sido generados en código simbólico multicuerpo (Vehicle Lisp), el cual proporciona mayor rapidez en cálculo al estar

muy optimizado. Los modelos físicos con los que se ha desarrollado esta herramienta han sido obtenidos y validados en más de 350 simulaciones de conducción por lo que le confiere una alta fidelidad al ser comparados con ensayos reales.

Es por todo lo anterior que este software está muy valorado en la industria del automóvil, tanto es así, que más de 30 fabricantes de automóviles (General Motors, Toyota, Honda, Ford, etc.), más de 60 proveedores, y más de 150 laboratorios de investigación y universidades lo utilizan.

Gracias a esta herramienta de simulación, los diseños de vehículos pueden ser probados y analizados antes de construirlos, con el objetivo de verificar que el comportamiento del mismo se ajusta a lo deseado. A esto hay que unirle que los tiempos de diseño como los costes se ven claramente reducidos.

Las principales aplicaciones del software CarSim son:

- ❖ Los ingenieros de pruebas simulan cientos de pruebas para identificar problemas en el diseño en el caso de que las hubiera.
- ❖ Los desarrolladores de controles avanzados (frenos, estabilidad, tracción, etc.) prueban sus diseños de control simulado.
- ❖ Los fabricantes de automóviles y proveedores comprueban el controlador del hardware real utilizando sistemas de tiempo real HIL.

Investigadores y otras personas utilizan los modelos matemáticos CarSim en simuladores de conducción, que van desde sistemas de bajo coste como un dispositivo de juego hasta simuladores de movimientos de gran escala y tamaño como el simulador de Toyota.

CarSim presenta un interfaz bastante intuitivo por lo que conociendo de manera general su funcionamiento se puede realizar numerosos test sin más que navegar por cada uno de sus menús y bibliotecas. En el Anexo II, Capítulo 15, se ha llevado a cabo un tutorial básico para conocer el software [23], [66], [78].

9.3 JAVA-NNS

Java Neural Network Simulator (Java-NNS) es un simulador de redes neuronales desarrollado en Wilhelm-Schickard - Instituto de Ciencias de la Computación- (WSI) en Tübingen, Alemania y que puede ser adquirido de manera libre a través de su página web.

Se basa en Stuttgart Neural Network Simulator (SNNS) 4.2 Kernel, con una nueva interfaz gráfica de usuario escrito en lenguaje Java. Como consecuencia, las capacidades de Java-NNS son sobre todo igual a las capacidades de los SNNS, mientras que la interfaz de usuario ha sido recientemente diseñada para que sea más fácil e intuitivo de usar. Algunas funciones utilizadas con frecuencia en SNNS (por ejemplo, visualización en tres dimensiones de las redes neuronales) se han pospuesto para futuras versiones de Java-NNS.

Una de las grandes ventajas que ofrece Java-NNS frente a SNNS es que es más independiente de la plataforma. SNNS fue creado en Unix mientras que Java-NNS puede ser ejecutado en sistemas operativos como Windows, RedHat de Linux, Solaris y MAC instalando únicamente el entorno de ejecución de Java.

Su antecesor, SNNS, cuenta con las siguientes características. Fue desarrollado en Unix en el Instituto de Sistemas de Alto Rendimiento Paralela y Distribuida (IPVR) de la Universidad de Stuttgart para servir de entorno de simulación eficiente y flexible para la investigación y aplicación de las redes neuronales.

El simulador SNNS consta de dos componentes principales:

- 1) Simulador Kernel escrito en C.
- 2) Interfaz gráfica de usuario bajo X11R4 o X11R5.

El simulador Kernel opera en las estructuras de datos de la red interna de las redes neuronales y lleva a cabo todas las operaciones de aprendizaje y memoria. Este software permite ser extendido por el usuario incluyendo nuevas funciones de activación, de salida, de página o nuevos procesos de aprendizaje. Debe ser escrito en lenguaje C y relacionado con el simulador Kernel.

Las siguientes arquitecturas de red y procedimientos de aprendizaje se incluyen:

- ❖ Backpropagation (BP) for feedforward networks
 - vanilla (online) BP
 - BP with momentum term and flat spot elimination
 - batch BP
- ❖ Counterpropagation

- ❖ Quickprop
- ❖ Backpercolation 1
- ❖ RProp
- ❖ Generalized radial basis functions (RBF)
- ❖ ART1
- ❖ ART2
- ❖ ARTMAP
- ❖ Cascade Correlation
- ❖ Recurrent Cascade Correlation
- ❖ Dynamic LVQ
- ❖ Backpropagation through time (for recurrent networks)
- ❖ Quickprop through time (for recurrent networks)
- ❖ Self-organizing maps (Kohonen maps)
- ❖ TDNN (time-delay networks) with Backpropagation
- ❖ Jordan networks
- ❖ Elman networks and extended hierarchical Elman networks
- ❖ Associative Memory

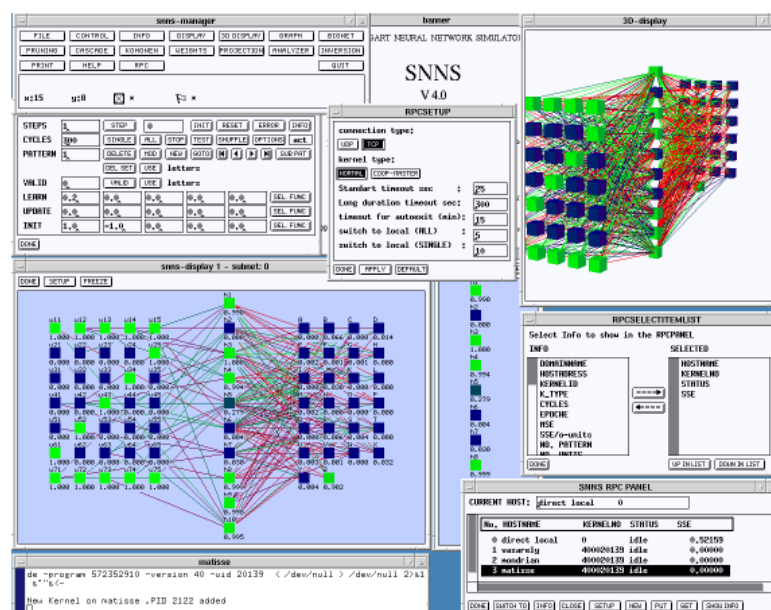


Figura 9.3-1: Java-NNS.

En el Anexo II, Capítulo 15, se ha dispuesto un tutorial básico para su utilización. Se ha hecho mayor hincapié en el método de aprendizaje Backpropagation por ser el que se ha usado en el estudio realizado en este proyecto fin de carrera.

10 PROPUESTA DE MODELO DE ESTIMACIÓN DEL ÁNGULO DE DERIVA



10.1 INTRODUCCIÓN

El propósito de este proyecto fin de carrera es el de comprobar la posibilidad de estimar el ángulo de deriva mediante el método de redes neuronales usando el software Java-NNS. Tal y como se comentó en el apartado 6.4, se trata de una de las líneas de investigación que se están llevando a cabo en los últimos tiempos debido a la evolución experimentada en los sistemas informáticos y a la capacidad que tienen estos sistemas para crear relaciones matemáticas no lineales entre distintas variables sin necesidad de modelos que las relacionen.

Para que un sistema de redes neuronales sea implementado en un vehículo de producción y que mediante las variables obtenidas por los sensores que llevan comúnmente instalados sea capaz de estimar el ángulo de deriva debe ser primeramente entrenada y validada. Durante este apartado lo que se va a realizar es la labor de entrenamiento y validación de diferentes redes propuestas con el fin de escoger la que estime de manera más eficiente y con mayor exactitud el ángulo de deriva. Para ello se han seguido las siguientes fases:

- ❖ Obtener los patrones.
- ❖ Tratar los datos.
- ❖ Crear y configurar las redes.
- ❖ Análisis de cada red.
- ❖ Comparativa de las redes.

10.2 OBTENER LOS PATRONES

Para que una red neuronal sea capaz de determinar los valores de las variables de salida, es necesario que adquiera experiencia realizando previamente un aprendizaje a partir de unos patrones de entrenamiento. Estos patrones de entrenamiento incluyen valores de las variables de entrada al sistema y los valores que deben obtenerse a la salida. De tal modo que para cada conjunto de datos de entrada, la red irá ajustando todos los parámetros (pesos y ganancias) para que los valores de salida del sistema sean lo más próximos a los valores de salida que se dieron en el patrón de entrenamiento y, así, al final del aprendizaje la red pueda dar una respuesta satisfactoria para cualquier set de datos de entrada.

Una vez realizado el proceso de entrenamiento de una red neuronal es necesario comprobar si esta es capaz de obtener por sí sola los valores de salida. Para ello, se hace uso de los patrones de validación. Estos patrones tienen el mismo formato que los de entrenamiento, unos valores de entrada y los valores de salida que la red debería estimar. En este caso, la red va introduciendo los conjuntos de datos de entrada para estimar las variables de salida y así calcular la desviación existente entre el valor de salida estimado y el proporcionado en el patrón de validación. Por tanto,

una red neuronal será validada cuando para cualquier set de valores de entrada es capaz de proporcionar unos valores de salida muy próximos a los reales.

Para conocer si una red es apta para dar soporte a un sistema es muy importante elegir bien los patrones de entrenamiento y validación de la misma, objetivo que se va a realizar en este apartado.

El presente proyecto fin de carrera estudia la posibilidad de obtener el ángulo de deriva a través de las redes neuronales por lo que esta será la variable de salida del sistema. Definir cuáles serán las variables de entrada es una tarea ardua ya que como se comentó en el Capítulo 5, el ángulo de deriva de un vehículo depende multitud de variables difícilmente conocidas a priori o medidas. Aún así, debido a dos motivos primordialmente se ha decidido que las variables de entrada sean: la aceleración lateral, la velocidad longitudinal, el ángulo de dirección y la velocidad de guiñada. Los dos motivos son:

- ❖ Son las variables más influyentes en el ángulo de deriva, tanto es así que la mayoría de los estudios previos que se encuentran en la bibliografía trabajan con estas variables. En [70] se demuestra mediante formulación cinemática como el ángulo de deriva depende esas cuatro variables y de los parámetros geométricos del vehículo.

$$\beta = \left(-ma_y + \frac{l_r C_r - l_f C_f}{v_x} r + C_f \delta \right) / (C_f + C_r) \quad \text{Ec.10.2-1}$$

donde

- m: masa del vehículo.
 - δ : ángulo de dirección.
 - V_x : velocidad longitudinal del vehículo.
 - a_y : aceleración lateral.
 - r : velocidad de guiñada.
 - l_f y l_r : distancia del CG al eje delantero y trasero respectivamente.
 - C_f y C_r : curva de rigidez delantera y trasera.
 - β : ángulo de deriva de un vehículo.
- ❖ Los vehículos hoy en día están equipados con sensores para dar soporte a los sistemas de seguridad que incorporan. Las variables de entrada deben poder ser obtenidas a través de esos sensores sin necesidad de instalar ninguno más.

Por tanto el sistema estará definido con las siguientes variables:

- ❖ Variables de entrada:
 - Aceleración lateral (a_y)
 - Velocidad longitudinal (V_x)
 - Ángulo de dirección (δ)
 - Velocidad de guiñada (ψ)
- ❖ Variable de salida:
 - Ángulo de deriva (β)

Los patrones de entrenamiento y de validación deberán contener para cada instante de tiempo las mencionadas variables de entrada y de salida. Existen dos maneras de obtener esos datos:

- ❖ Primer método: consiste en realizar diferentes maniobras con un vehículo real totalmente equipado de sensores (sensores GPS, acelerómetros, odómetros, giróscopos, sensor de ángulo de dirección, sensores ópticos y sensores dinamométricos) e ir registrando todos los parámetros en cada instante de tiempo. Cuyas ventajas y desventajas son las siguientes:
 - Ventajas:
 - Se trata de ensayos reales por lo que los datos obtenidos serán representativos de lo que se obtendrán cuando el sistema esté implantado en los vehículos de producción.
 - Desventajas:
 - El coste de los ensayos es elevado ya que es necesario equipar un vehículo con los sensores, en especial el sistema que obtiene el valor real del ángulo de deriva.
 - El tiempo necesario para obtener diferentes ensayos es elevado.
 - No se puede tener el control exigido sobre los parámetros durante los ensayos como la velocidad, coeficiente de adherencia...
- ❖ Segundo método: consiste en simular las maniobras mediante programas informáticos. Cuyas ventajas y desventajas son las siguientes:
 - Ventajas:
 - Permite realizar todas las simulaciones que uno desee sin que el coste se vea incrementado exponencialmente.
 - Permite tener control sobre todos los parámetros de las maniobras.
 - Permite comparar el comportamiento del vehículo bajo diferentes condiciones.
 - Desventajas:
 - Aunque los softwares están muy evolucionados no llegan a ser sistemas reales por lo que es necesario conocer y tener en cuenta sus limitaciones.

Entre las dos opciones, se ha elegido para el estudio de este proyecto la segunda ya que no se disponía de un vehículo equipado con los sensores necesarios y, en cambio, sí de un software de simulación que satisface los requisitos exigidos para este proyecto, CarSim versión 8.02. Ver apartado 9.2.

Un punto importante del trabajo fue decidir que maniobras serían reproducidas para obtener unos patrones de entrenamiento y validación representativos de las posibles condiciones en las que se puede desenvolver un vehículo. Debido a la no lineal que presenta el ángulo de deriva (ver Capítulo 5), es necesario que las maniobras deban estar caracterizadas de acuerdo a obtener comportamientos del vehículo dentro de la zona lineal y de la zona no lineal con el fin de tener suficiente información a la hora de entrenar la red. Por ello, las maniobras deben tener en cuenta las siguientes consideraciones: [68]

- ❖ La red debe ser entrenada con maniobras en sentido horario y antihorario para que no se produzca un comportamiento asimétrico y cuando se analicen otras maniobras no incluidas en los patrones de entrenamiento ofrezcan valores correctos.
- ❖ Se han de tener en cuenta maniobras con coeficientes de adherencia altos y bajos para simular el asfalto seco y superficies de baja adherencia (agua, nieve, hielo...). De este modo la red podrá predecir los valores del ángulo de deriva sea cual sea la condición de adherencia.
- ❖ Deben encontrarse situaciones en el que la dinámica del vehículo sea lineal y no lineal.

10.2.1 DEFINICIÓN DE LAS MANIOBRAS

El modo de obtener los patrones de entrenamiento y validación ha sido diferente. A continuación se detallan:

10.2.1.1 Patrones de entrenamiento

En la literatura se encuentran diferentes maniobras realizadas para validar los datos que habían obtenido en sus estudios o para obtener los valores con los que realizarlos. Las maniobras y parámetros que son tenidos en consideración son los siguientes:

- ❖ Tipos de maniobra:
 - Cambio de carril.
 - Doble cambio de carril.
 - Ángulo de dirección constante.
 - Ángulo de dirección sinusoidal.
 - Trayectoria en curva.

❖ Parámetros que se han considerado:

- Velocidad longitudinal.
- Coeficiente de adherencia.
- Aceleración y deceleración longitudinal.
- Peralte de la carretera.
- Inclinación de la carretera.
- Características de los neumáticos.

A continuación se muestran las diferentes maniobras que se llevaron a cabo en otros estudios previos a este:

❖ En el estudio [33] se realizaron dos tipos de test mostrados en la Figura 10.2.1.1-1.

- El primero era un slalom en el que en una primera fase el vehículo aceleraba longitudinalmente alcanzando los 0.3g, en una segunda fase realizaba el slalom con aceleraciones laterales contenidas en $-0.6 \text{ g} < a_y < 0.6 \text{ g}$ y en una tercera fase desaceleraba longitudinalmente alcanzando los -0.5 g.
- La segunda maniobra consistía en describir una rotonda con un ángulo de dirección menor de 11° .

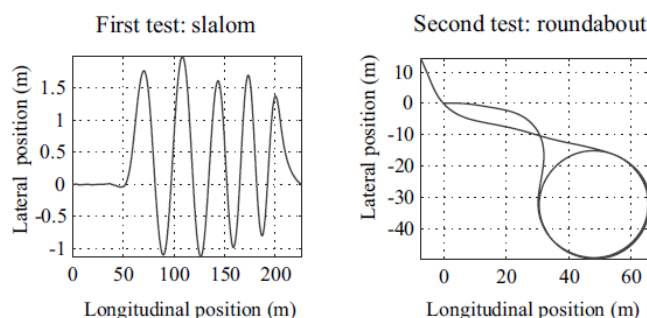


Figura 10.2.1.1-1: Test realizados en [33].

❖ En la Tabla 10.2.1.1-1 se recogen las maniobras realizadas en [70], fueron simuladas mediante el software CarSim.

Tabla 10.2.1.1-1: Test realizados en [70].

Tipo maniobra	Peralte [deg]	Vx [Km/h]	μ [-]
Doble cambio de carril	0	120	1.1
Doble cambio de carril	10-5%	40	0.35
Dirección sinusoidal	0	30	0.15
Dirección sinusoidal	10%	80	0.9
Dirección sinusoidal	10%	40	0.35

- ❖ En la Tabla 10.2.1.1-2 se recogen las maniobras realizadas en [59].

Tabla 10.2.1.1-2: Test realizados en [59].

Tipo maniobra	dirección [deg]	Vx [Km/h]
Dirección constante	30	100
Dirección constante	80	100
Doble cambio de carril (ISO)	-	80

- ❖ En [36] realizaron las siguientes maniobras recogidas en la Tabla 10.2.1.1-3.

Tabla 10.2.1.1-3: Test realizados en [36].

Tipo maniobra	Vx [Km/h]	μ [-]
Doble cambio de carril	100	0.75 (asfalto seco)
Doble cambio de carril	100	0.5 (asfalto mojado)
Cambio de sentido	80	0.75 (asfalto seco)

- ❖ En el estudio [40] se realizaron tres tipos de maniobras diferentes.
- La primera de ellas consistió en hacer girar el vehículo en una curva de 40 m de radio e ir incrementando la velocidad hasta que el comportamiento del vehículo fuera inestable. La máxima aceleración lateral obtenida fue 0.93 g cuando el vehículo montaba neumáticos de verano y 0.84 g con los de invierno.
 - El segundo ensayo realizado fue variando el ángulo de dirección a una velocidad de 200Km/h, logrando aceleraciones laterales de $\pm 0.4g$.
 - Por último, un doble cambio de carril (ISO) a 120 Km/h con un peralte de 0.5°. Obtuvieron aceleraciones laterales de -0.9g.
- ❖ En [68] ha sido realizado el estudio mediante redes neuronales por lo que se han tomado como referencia las maniobras que en él se han establecido. Las maniobras seleccionadas obtienen comportamientos lineales y no lineales, llegando a alcanzar aceleraciones cercanas a la aceleración lateral máxima (1g). Las maniobras han sido dispuestas en la Tabla 10.2.1.1-4.

Tabla 10.2.1.1-4: Test realizados en [68].

Tipo maniobra	dirección [deg]	V_x [Km/h]	μ [-]
Dirección constante	±45	30	1
Dirección constante	±45	100	1
Dirección constante	±45	100	0.3
Dirección constante	±100	100	1
Dirección constante	±80	80	1
Dirección constante	±80	80	0.3
Dirección sinusoidal	±30	30→100	1

Por tanto, las maniobras que se han llevado a cabo en este estudio están basadas en las realizadas en [68]. Son del tipo dirección constante. En cada maniobra se irán variando los siguientes tres parámetros:

- ❖ Coeficiente de adherencia entre el neumático y calzada.
- ❖ Velocidad longitudinal a la que se desplaza el vehículo.
- ❖ Ángulo que se ha girado el volante del automóvil.

De acuerdo a satisfacer las tres consideraciones que se recogieron anteriormente, los valores que han tomado los parámetros son los mostrados en la Tabla 10.2.1.1-5:

Tabla 10.2.1.1-5: Valores que toma cada variable en las distintas maniobras.

PARÁMETRO	VALORES
Coef. de adherencia [-]	0.3 1
Velocidad longitudinal [Km/h]	30 65 100 130
Ángulo de dirección [deg]	±45 ±75 ±100 ±125 ±150

Modificando tan solo esos tres parámetros y manteniendo invariables el resto de parámetros se han reproducido un total de 64 maniobras, listadas en la Tabla 10.2.1.1-6.

Tabla 10.2.1.1-6: Maniobras simuladas.

Maniobras	C. Fricción [-]	Velocidad [Km/h]	Dirección [deg]	Maniobras	C. Fricción [-]	Velocidad [Km/h]	Dirección [deg]
1, 2	0,3	30	± 45	33, 34	1	30	± 150
3, 4	0,3	30	± 100	35, 36	1	65	± 45
5, 6	0,3	30	± 150	37, 38	1	65	± 75
7, 8	0,3	65	± 45	39, 40	1	65	± 100
9, 10	0,3	65	± 100	41, 42	1	65	± 125
11, 12	0,3	65	± 150	43, 44	1	65	± 150
13, 14	0,3	100	± 45	45, 46	1	100	± 45
15, 16	0,3	100	± 100	47, 48	1	100	± 75
17, 18	0,3	100	± 150	49, 50	1	100	± 100
19, 20	0,3	130	± 45	51, 52	1	100	± 125
21, 22	0,3	130	± 100	53, 54	1	100	± 150
23, 24	0,3	130	± 150	55, 56	1	130	± 45
25, 26	1	30	± 45	57, 58	1	130	± 75
27, 28	1	30	± 75	59, 60	1	130	± 100
29, 30	1	30	± 100	61, 62	1	130	± 125
31, 32	1	30	± 125	63, 64	1	130	± 150

10.2.1.2 Patrones de validación

En cambio, los patrones de validación se han obtenido a través de una única simulación. Ha consistido en simular una posible situación de un vehículo que se desplaza por las calles y carreteras de una ciudad, para ello, se ha hecho uso del circuito ya definido en la biblioteca del programa CarSim, ver Figura 10.2.1.2-1. Tan solo se han modificando las velocidades, ángulos de dirección y coeficiente de adherencia para conseguir que el vehículo trace la trayectoria deseada sin que se salga de la calzada y creando situaciones de comportamiento lineal y no lineal.

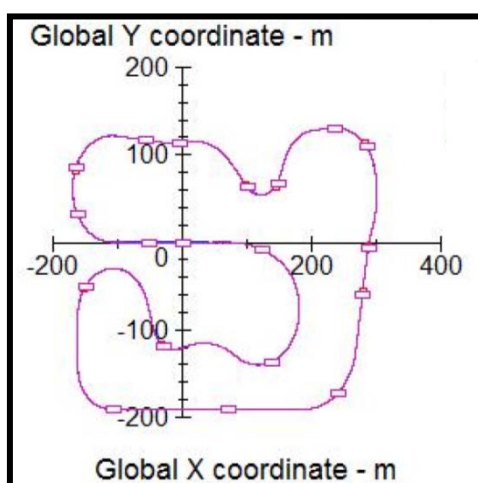


Figura 10.2.1.2-1: Circuito para patrones de validación.

A continuación se explica cómo han sido configuradas las maniobras en el Software CarSim. Para más información ir al Anexo II, Capítulo 15, donde se ha dispuesto un tutorial sobre este programa.

10.2.2 CONFIGURACIÓN DE LAS MANIOBRAS EN CARSIM

Se han configurado los siguientes tres puntos en CarSim para cada maniobra:

- ❖ Configuración del vehículo
- ❖ Configuración de la maniobra
- ❖ Configuración de los parámetros a extraer

Puesto que todas las maniobras son iguales pero variando únicamente los valores del coeficiente de fricción, la velocidad y el ángulo de dirección, se va a explicar a modo de ejemplo la configuración de la maniobra 1 cuyos valores son los mostrados en la Tabla 10.2.2-1.

Tabla 10.2.2-1: Parámetros maniobra 1.

Maniobras	C. Fricción [-]	Velocidad [Km/h]	Dirección [deg]
1	0,3	30	+45

10.2.2.1 Configuración del vehículo

Un vehículo clase D sedán, Figura 10.2.2.1-1, ha sido utilizado para realizar las maniobras en CarSim. Un vehículo común en las calles de las ciudades aunque no tiene especial relevancia en el estudio ya que serviría cualquier otro.



Figura 10.2.2.1-1: Vehículo clase D sedán utilizado en las maniobras en CarSim.

10.2.2.2 Configuración de la maniobra

En la Figura 10.2.2.2-1 se muestra la ventana de CarSim desde la que se accede a todos los menús donde se configuran las variables de la maniobra. En esta misma ventana, se pueden configurar en el campo (A) la velocidad a la que se va a desplazar el vehículo. También, el tiempo de simulación de la maniobra en el campo (B). Se ha establecido para todas las maniobras un tiempo fijo, suficiente para registrar la cantidad necesaria de datos sobre el comportamiento del vehículo.

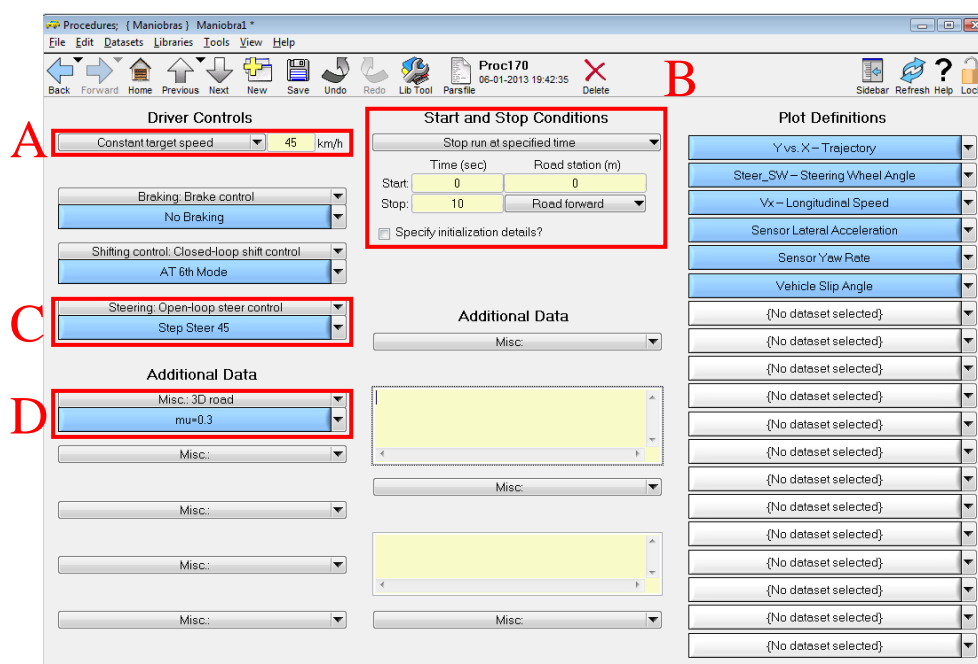


Figura 10.2.2.2-1: Menú configuración de la maniobra en CarSim.

Para configurar el ángulo de dirección en cada maniobra es necesario acceder al sub menú del campo (C), mostrado en la Figura 10.2.2.2-1. Para definir este parámetro se hace uso primeramente de la tabla de la derecha en la que se irán introduciendo tantas filas como sea necesario y los valores que debe tomar el ángulo de dirección (componente y) para cada instante de tiempo (componente x). En este caso, puesto que “ δ ” es constante durante toda la maniobra, se han creado dos filas, una para el instante inicial y otra para el tiempo final. Para la maniobra 1 será de 45°. Una vez rellenada la tabla se da al botón calcular para actualizar la gráfica de la ventana. Ver Figura 10.2.2.2-2.

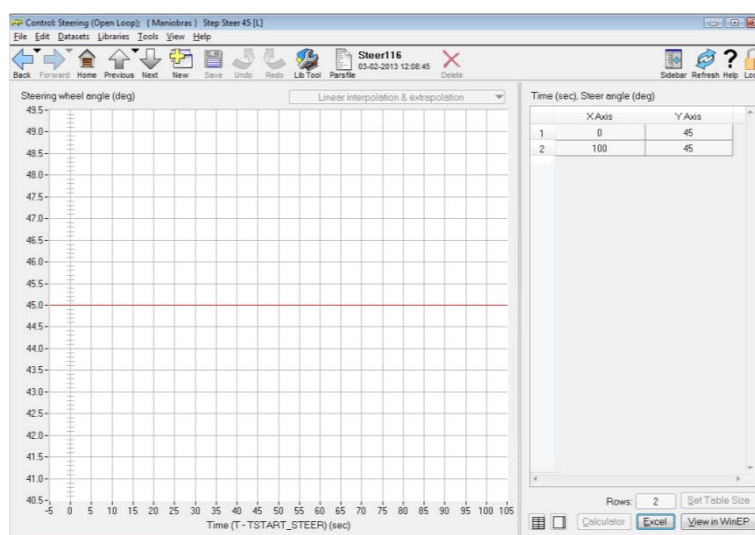


Figura 10.2.2.2-2: Configuración del ángulo de dirección.

Para configurar el coeficiente de adherencia así como el entorno en el que se va a desarrollar la maniobra es necesario acceder al sub menú del campo (D), mostrado en la Figura 10.2.2.2-1. En

la Figura 10.2.2.2-3 se muestran los dos campos, (E) y (F), que se van a modificar. En el campo (F) se configura el valor del coeficiente de adherencia mientras que en el campo (E) se configura la pista. Para el caso de los patrones de entrenamiento, se ha configurado una pista horizontal y rectangular mientras que para los patrones de entrenamiento se ha escogido el circuito de la biblioteca de CarSim, mostrado en la Figura 10.2.1.2-1.

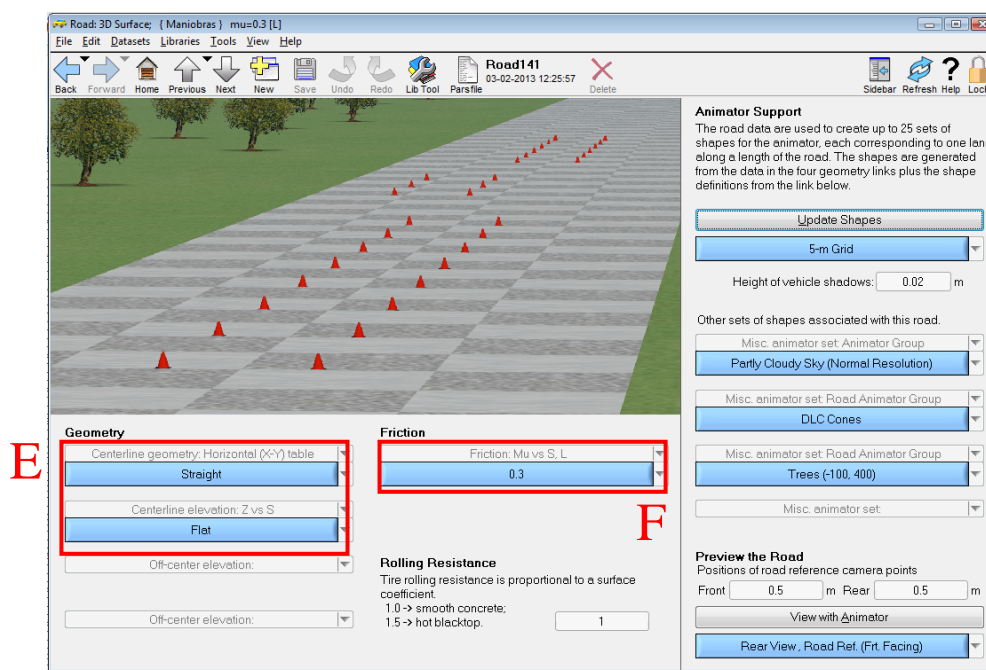


Figura 10.2.2.2-3: Configuración del entorno.

10.2.2.3 Configuración de los parámetros a extraer

Los sensores que van instalados en los vehículos están influenciados por ruido a causa de vibraciones, interferencias, la propia electrónica... Por lo que obtener en las simulaciones de las maniobras los valores reales de cada una de las variables no es interesante ya que el sistema se enfrentará a otro tipo de datos en la vida real. Por ello, es mejor idea, si existe la posibilidad, realizar las maniobras simulando que los valores son obtenidos a través de sensores reales y no a través de sensores ideales. Para conseguir esta característica, CarSim incorpora ruido blanco a los valores obtenidos, hipótesis que se ajusta muy bien a la realidad.

En la ventana mostrada en la Figura 10.2.2.3-1 se seleccionan los parámetros que se quieren estudiar al finalizar la simulación de la maniobra. Se han seleccionado seis parámetros: las cuatro variables de entrada (“Vx”, “ay”, “δ” y “ψ”), la variable de salida (“β”) y, a modo de interés visual, la trayectoria que ha seguido el vehículo para los parámetros definidos en cada maniobra. En todas aquellas variables en las que se podía definir la opción de sensor real se ha hecho.

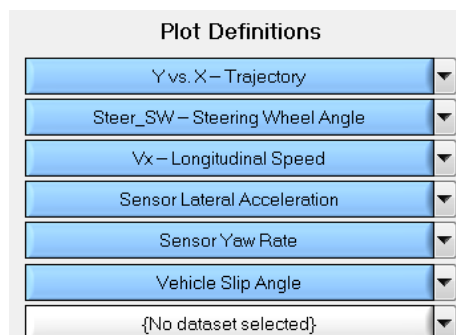


Figura 10.2.2.3-1: Parámetros seleccionados.

Una vez configurados los tres puntos marcados se puede volver a la ventana principal de CarSim y correr la simulación. Cuando haya finalizado se pueden visualizar los parámetros pedidos pulsando el botón “plot”, apareciendo en una ventana emergente las gráficas de cada variable en función del tiempo, ver Figura 10.2.2.3-2. De esta manera uno puede comprender de manera más o menos sencilla cual ha sido el comportamiento del vehículo bajo los parámetros que se le asignaron durante la configuración de la maniobra.

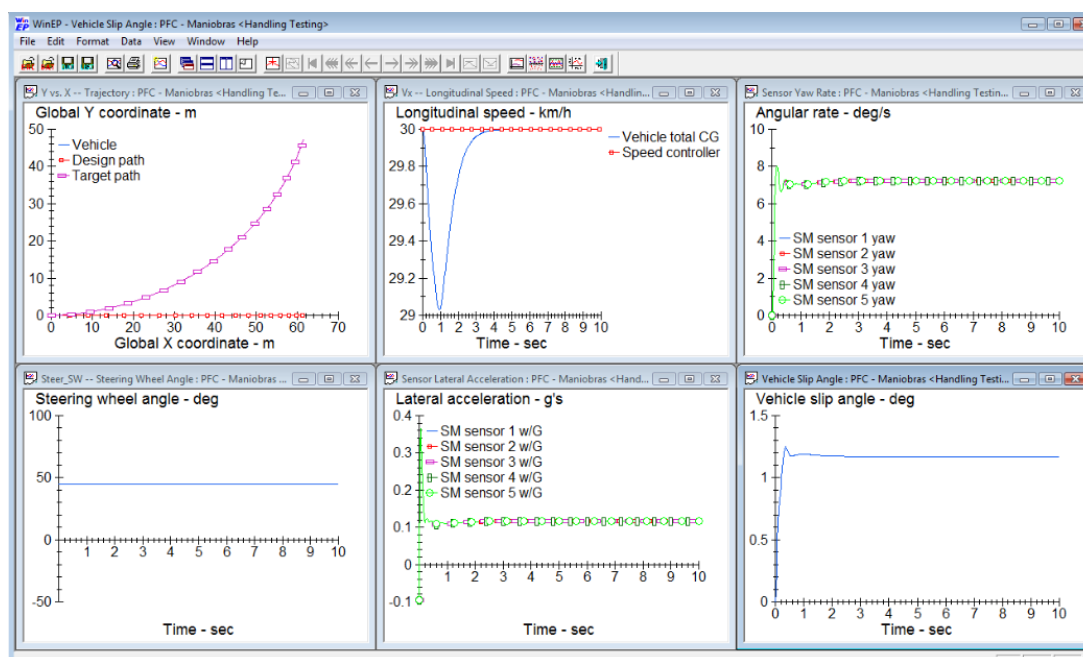


Figura 10.2.2.3-2: Gráficas obtenidas de la simulación de la maniobra 1.

Además, estas gráficas pueden ser guardadas en documentos de texto de tal modo que en ellos aparecerán dos columnas siendo la primera columna relativa al eje x de la gráfica, el tiempo en este caso, y en la segunda columna se visualizan los valores que ha ido tomando la variable seleccionada para cada instante de tiempo. Ver Figura 10.2.2.3-3.

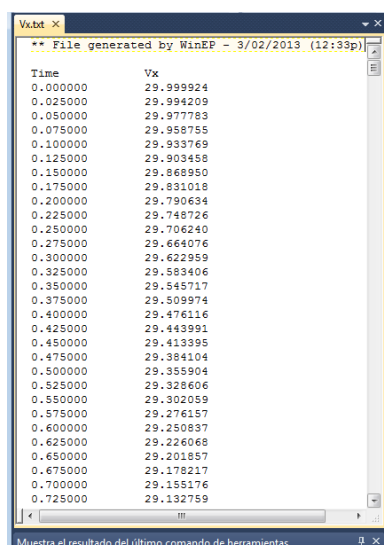


Figura 10.2.2.3-3: Valores de la velocidad en la maniobra 1 para cada instante de tiempo.

10.3 TRATAR LOS DATOS

Los patrones de entrenamiento y validación deben tener el formato con el que trabaja el programa Java-NNS. El documento debe contar con un encabezado en el que se resume el número de variables de entrada, de salida y el número total de patrones que dispone el documento. A continuación, aparecen en columnas, separadas por un espacio, los valores que va tomando cada variable para el mismo instante de tiempo en cada una de las maniobras realizadas. Las primeras columnas se refieren a las variables de entrada, las cuales se han dispuesto en el siguiente orden: “Vx”, “ ψ ”, “ δ ” y “ay” y la última columna a la variable de salida, “ β ”. De esta forma si se lee una línea se puede interpretar que para ese conjunto de datos de entrada, primeras cuatro columnas, se obtiene el valor registrado en la última columna. Ver Figura 10.3-1.

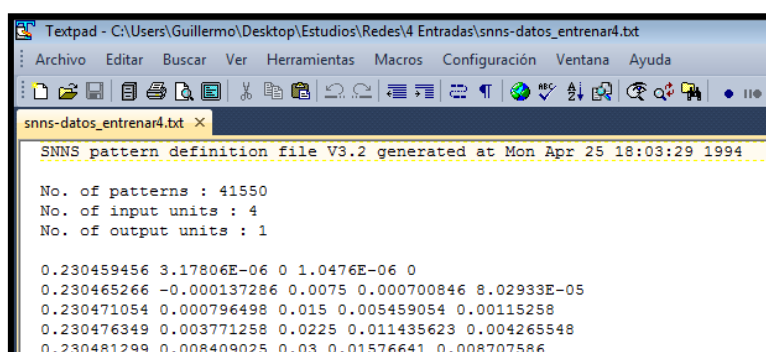


Figura 10.3-1: Encabezado del documento de patrones.

Además, hay que tener en cuenta otra cosa más aparte del formato a la hora de crear los patrones de entrenamiento y validación. Los valores registrados en cada variable no son del mismo orden de magnitud por lo que si se introdujesen así los datos en el programa Java-NNS no se conseguiría un buen estudio ya que habría variables que tendrían más peso que otras cuando el

software fuese ajustando los parámetros de la red (pesos y ganancias). Para solventar ese problema lo que se hace es normalizar las variables.

Debido a la gran cantidad de archivos que se han obtenido, cinco por cada maniobra haciendo un total de 325 archivos, se ha hecho uso del lenguaje de programación Python para automatizar parte del proceso del adecuación de los datos extraídos de CarSim y así generar los patrones de entrenamiento y validación de acuerdo al formato establecido por el software java-NNS y posteriormente de Excel. Estos programas tenían las siguientes funciones:

- ❖ Crear un único archivo por cada maniobra que registrara los valores de las cinco variables extraídas.
- ❖ Crear dos archivos, uno para entrenamiento y otro para la validación, con todos los valores de todas las maniobras, colocando cada una debajo de la anterior.
- ❖ Reemplazar los espacios entre columnas por tabulaciones ya que posteriormente es necesario importar el archivo en Excel para normalizar los valores.
- ❖ Cambiar los puntos decimales de los números por comas decimales.

Una vez agrupados todos los valores en un único archivo, tanto para entrenamiento como para validación, es necesario normalizar los valores y para ello se han importado los archivos de texto en el programa Excel. Se dividieron todos los valores de cada variable con el valor máximo absoluto encontrado tanto en el documento de entrenamiento como en el de validación. De esta manera, los variables que toman valores positivos y negativos (“ ψ ”, “ δ ”, “ a_y ” y “ β ”) estarán ahora comprendidos entre -1 y 1 mientras que las variables que solo toman valores positivos (“ V_x ”, “ μ ”) entre 0 y 1. El valor máximo obtenido para el ángulo de deriva y con el que se ha normalizado esta variable es 3.898210°, mostrado en la Tabla 10.3-1. Dicho valor es mayor que 2° lo que significa que en las maniobras que se han simulado en CarSim se han obtenido situaciones donde el vehículo tiene un comportamiento no-lineal, ver apartado 5.2.

Tabla 10.3-1: Valor β_{\max} con el que se normaliza

$$\beta_{\text{MAX}} = 3,898210^\circ$$

Finalmente se guardaron ambos documentos en formato texto manteniendo un solo espacio entre columnas y se añadió a cada documento el encabezado.

Al final de todo el proceso se han obtenido 41550 patrones de entrenamiento y 5251 patrones de validación. En la Figura 10.3-2 se muestra el aspecto de los patrones de entrenamiento y validación.

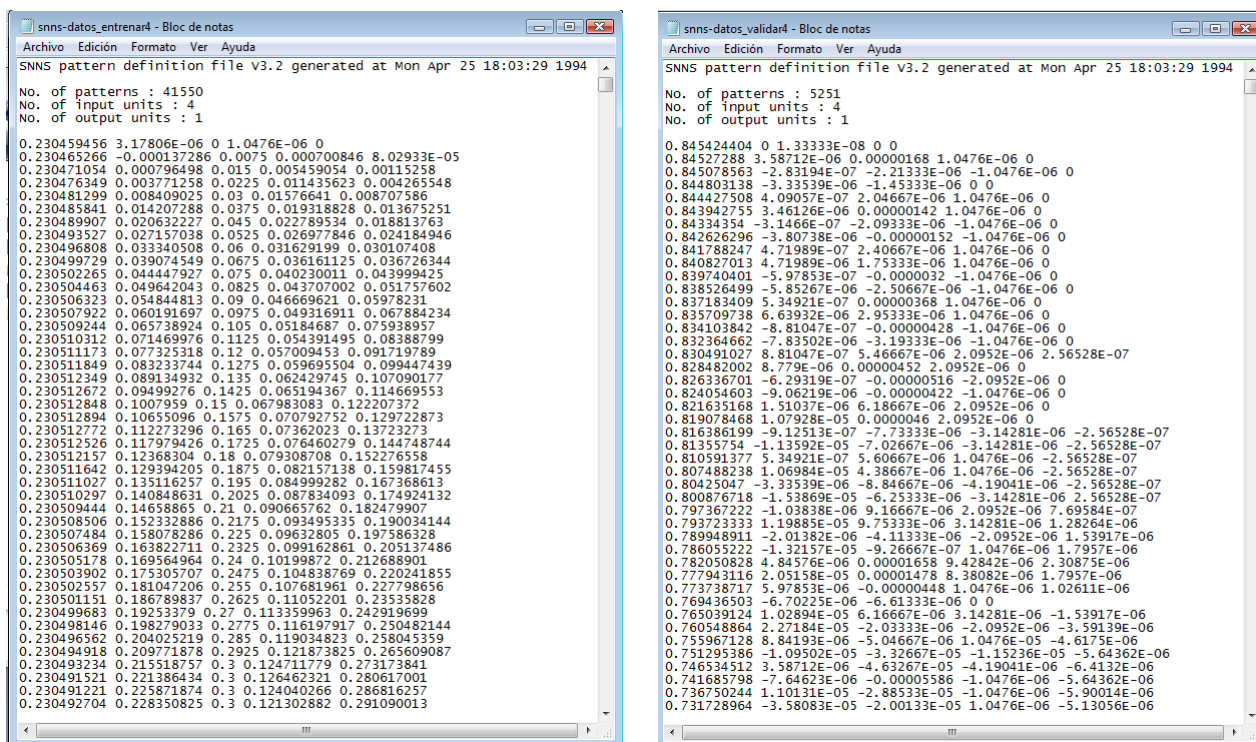


Figura 10.3-2: Patrones de entrenamiento (izda.) y validación (dcha.).

10.4 CREAR Y CONFIGURAR LAS REDES

Una vez generados los patrones de entrenamiento y validación es necesario decidir y definir cuáles serán las estructuras de las redes que se van a analizar. En el Anexo II, Capítulo 15, se ha dispuesto un tutorial básico sobre Java-NNS para crear y configurar las redes neuronales.

Las redes neuronales propuestas constan de:

- ❖ **Capa de entrada:** Solo contarán con una única capa de entrada formada por cuatro neuronas que corresponden a las variables de entrada del sistema en el siguiente orden: Velocidad longitudinal, velocidad de guiñada, ángulo de dirección y aceleración lateral.
- ❖ **Capa oculta:** Estudios previos han determinado que los resultados obtenidos del sistema no dependen tanto del número de capas ocultas y del número de neuronas que las formen sino del número total de neuronas que haya en las capas ocultas. Por tanto, dos redes con el mismo número total de neuronas en las capas ocultas dispuestas en una, dos o n-capas ocultas arrojarán el mismo valor de salida. Por este motivo, solo se han propuesto redes de una y dos capas ocultas. El objetivo de crear redes de dos capas ocultas no es otro que comprobar esa realidad. Las distintas redes propuestas se diferencian únicamente en el número de capas ocultas y en el número de neuronas que las conforman.

- ❖ **Capa de salida:** Solo contarán con una única capa de salida formada por una neurona, la cual corresponde a la variable de salida, el parámetro que se quiere estimar: ángulo de deriva del vehículo.

Se ha definido la siguiente nomenclatura para distinguir las tipologías de red creadas: **Red (a)-(b)-(c)**, donde (a) es el número de neuronas en la capa de entrada, (b) corresponde a las capas ocultas, en caso de haber más de una se mostrará el número de neuronas de cada capa de la siguiente manera (b1-b2), y (c) es el número de neuronas de la capa de salida. De este modo la red (4)-(5)-(1) estará formada por cuatro neuronas en la capa de entrada, una única capa oculta con cinco neuronas y una capa de salida con una neurona. En cambio, la red (4)-(5-5)-(1) estará formada por cuatro neuronas en la capa de entrada, dos capas ocultas con cinco neuronas cada una y una capa de salida con una neurona. Ver Figura 10.4-2.

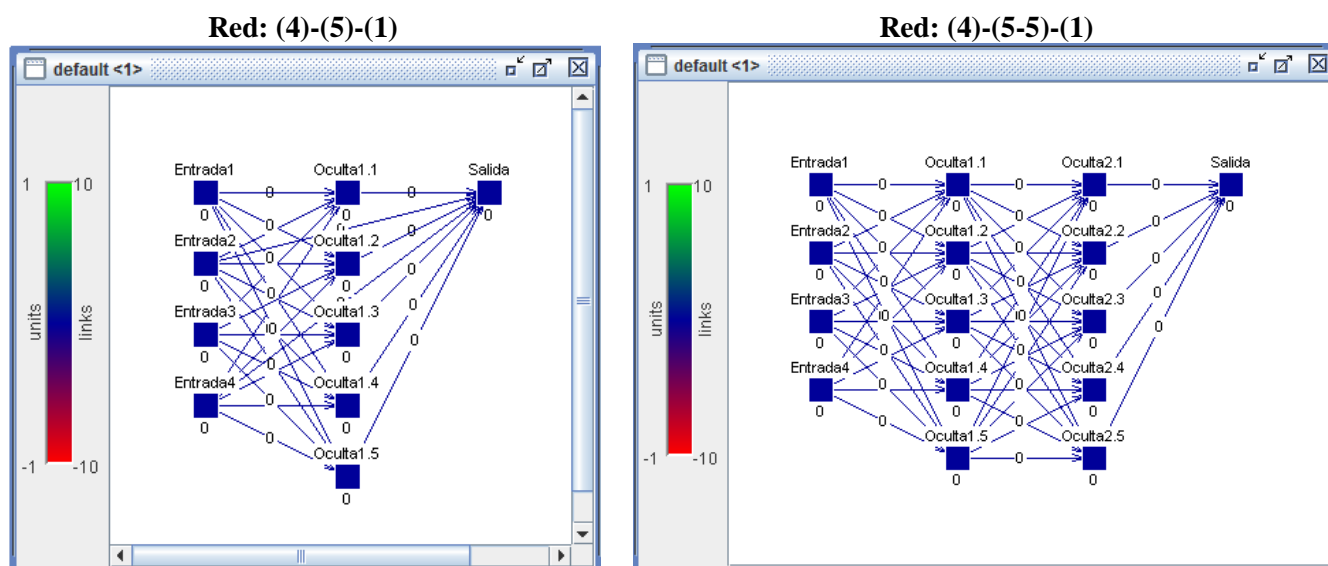


Figura 10.4-1: Nomenclatura estructura redes.

Las estructuras de redes que se han definido son las siguientes:

- ❖ Red1: (4)-(5)-(1)
- ❖ Red2: (4)-(10)-(1)
- ❖ Red3: (4)-(15)-(1)
- ❖ Red4: (4)-(20)-(1)
- ❖ Red5: (4)-(25)-(1)
- ❖ Red6: (4)-(5-5)-(1)
- ❖ Red7: (4)-(10-10)-(1)

Además de la tipología hay que definir cuál será la función de activación, la función de salida en cada neurona y el modo en el que se unirán las neuronas entre sí para crear la red.

- ❖ **Función de activación:** es la característica principal de una neurona. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total a la misma, ver apartado 7.4.2. Las capas de entrada y oculta han sido definidas mediante la función tangente hiperbólica, puesto que presenta una buena respuesta al tipo de función y a que los datos que se van a manejar han sido normalizados entre los valores -1 y 1. En cambio, la capa de salida ha sido definida mediante la función identidad ya que se quiere conocer el valor que obtiene la red a la salida sin que se vea modificado por ninguna función. Ver Figura 10.4-2.
- ❖ **Función de salida:** La función de salida convierte el estado de la neurona en la salida hacia la siguiente neurona que se transmite por la sinapsis. Se ha tomado la función identidad para todas las neuronas. De esta manera, la señal de salida de cada neurona coincide con el propio estado de activación de la neurona. Ver Figura 10.4-2.
- ❖ **Conexiones:** Cada una de las neuronas de una capa ha sido unida al resto de neuronas de la siguiente capa.

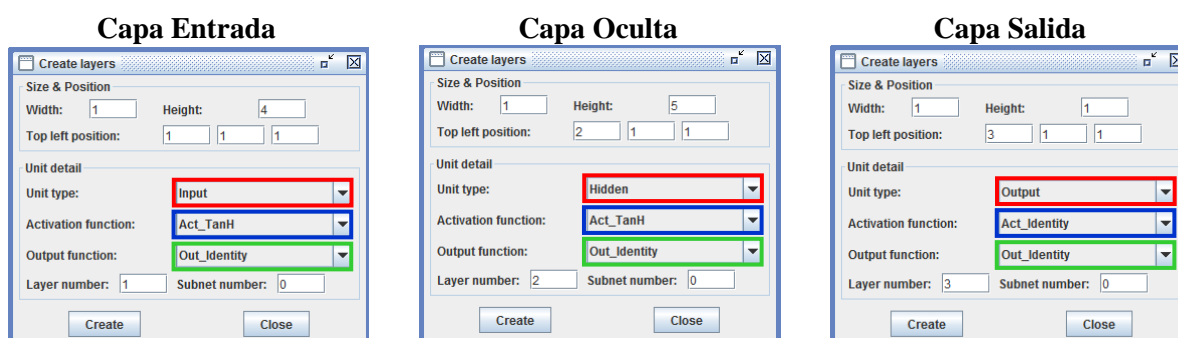


Figura 10.4-2: Nomenclatura estructura redes. En rojo se define el tipo de capa, en azul la función de activación y en verde la función de salida de todas las neuronas de cada capa.

Debido a que primeramente solo se realizará el entrenamiento de la red solo será necesario cargar el documento de los patrones de entrenamiento. Una vez haya sido entrenada, se cargará el documento con los patrones de validación y se realizará un análisis para comprobar si la red es capaz de obtener los valores del ángulo de deriva para cualquier set de datos de entrada.

La función de aprendizaje que se ha elegido para el aprendizaje de todas las redes neuronales ha sido Backpropagation. En dicha función es necesario introducir dos valores:

- ❖ **Error (dmax):** Valor a partir del cual el error ya no es retropropagado. El valor definido es 0.1.
- ❖ **Tasa de aprendizaje (η):** elemento encargado de la velocidad de aprendizaje ya que es el que define el incremento de los pesos para avanzar a través de la superficie de error definida por la técnica de gradiente descendiente, en la que se basa Backpropagation, y encontrar el mínimo error. De este modo, se introducirán valores

elevados para acercarse lo más rápido posible al error y se irá reduciendo su valor para ir aproximándose poco a poco hasta que el algoritmo converja según se muestra en la Figura 10.4-3, ver apartado 8.2. Los valores que se han ido introduciendo son los siguientes: 0.2, 0.1, 0.02, 0.01 y 0.005, a partir de este último valor no se obtiene una mejora relevante en los resultados. Se fue modificando su valor cuando la gráfica de error de mantenía constante, lo cual significaba que el algoritmo no era capaz de encontrar el mínimo puesto que saltaba entre las dos ramas de la función que llegan al mínimo. Ver Figura 10.4-4.

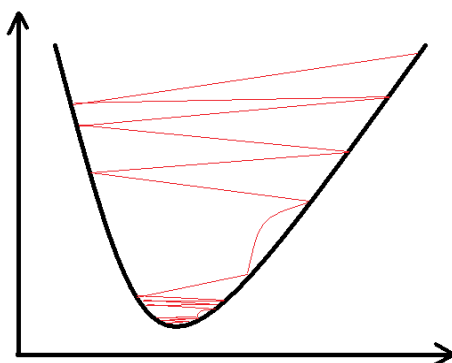


Figura 10.4-4: Representación de cómo recorre la superficie de error al variar la tasa de aprendizaje.

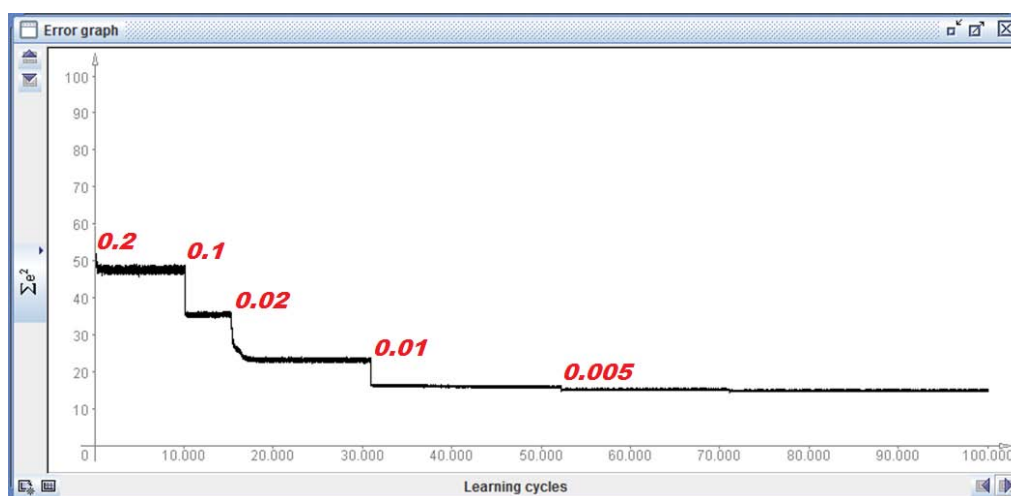


Figura 10.4-4: Variación de la tasa de aprendizaje según la evolución del error.

El número de ciclos que se ha definido para la realización del aprendizaje ha variado según aprendía la red. En cada red se especificará su valor.

Todos esos parámetros han sido introducidos al programa en el Panel de control. Para más información consultar el apartado 15.2.4.

Antes de presionar sobre el botón Learn Current o All es necesario extraer todas las ventanas donde se irán registrando los datos que son de interés para el estudio: el gráfico de errores y de pesos, la ventana de texto (.log) además de mantener la ventana donde está representada la red. En

la Figura 10.4-5 se muestran. En la ventana de gráfico de errores es necesario mantener seleccionada la opción $1/n\Sigma e^2$ para que se vayan registrando en la ventana de texto (.log) los valores de error en formato MSE.

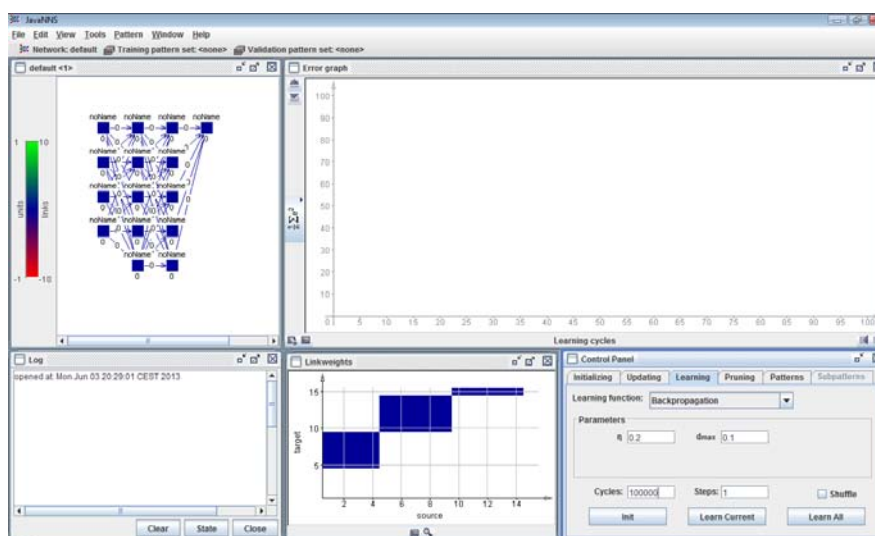


Figura 10.4-5: Pantalla entrenamiento redes.

Al finalizar el entrenamiento, se abre la ventana mostrada en la Figura 10.4-6 y se configura de acuerdo a lo comentado en el apartado 15.2.8. Primeramente es necesario cargar el documento con los patrones de validación con los que se analizará si la red es apta.

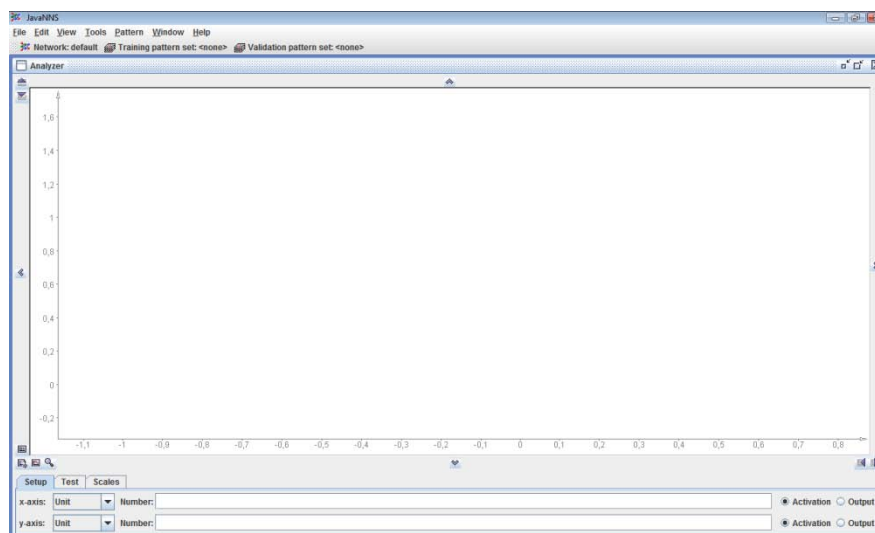


Figura 10.4-6: Pantalla validación de las redes.

Para el tratamiento de los archivos obtenidos del entrenamiento y validación de las redes se ha hecho uso de programas escritos en Python.

10.5 ANÁLISIS DE CADA RED

En este apartado se muestran los resultados obtenidos de los procesos de entrenamiento y validación para cada una de las redes propuestas.

El esquema seguido en cada red es el siguiente:

- ❖ Primeramente, se expone como ha ido evolucionando el error cuadrático medio hasta estabilizarse y el número de ciclos empleados para ello.
- ❖ En segundo lugar, se presenta un breve análisis de los gráficos obtenidos, tanto para el entrenamiento como para la validación, en los que se representan los valores normalizados (entre -1 y 1) del ángulo de deriva estimados por la red (eje de coordenadas), y los valores reales obtenidos mediante las maniobras simuladas en CarSim e introducidas en Java-NNS a través de los patrones respectivos (eje de abscisas). Además se dispone de una Tabla en la que se han recogido los siguientes datos:
 - Error Máx: Mayor desviación el ángulo de deriva estimado y real.
 - Error Medio: Media de las desviaciones entre en ángulo de deriva estimado y real.

10.5.1 Red 1: (4)-(5)-(1)

La Red 1 fue entrenada durante 100000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0091 registrado en los 10000 ciclos a 0.0028 al final del aprendizaje, datos recogidos en la Tabla 10.5.1-1. Además, se puede apreciar en la Figura 10.5.1-1 cómo ha ido evolucionando el error.

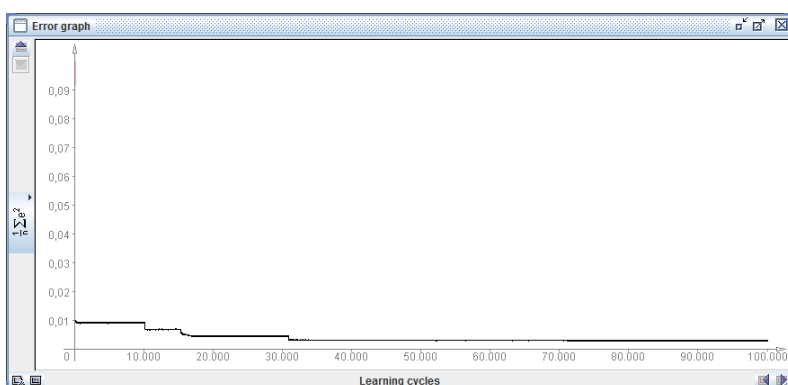


Figura 10.5.1-1: Evolución del MSE de la Red 1: (4)-(5)-(1)

Tabla 10.5.1-1: MSE Red 1: (4)-(5)-(1)

Step	MSE
10000	0.009155397663977322
20000	0.0044169842776878064
30000	0.0043635062321416085
40000	0.003022165576336929
50000	0.003029228323778637
60000	0.002922246261361303
70000	0.002886133406462385
80000	0.0028038727762675973
90000	0.0028226556743446746
100000	0.0028417534974388248

La red 1 no logra ajustarse de manera satisfactoria a los valores reales del ángulo de deriva durante el entrenamiento. Sin embargo, los valores estimados en la fase de validación se ajustan bastante bien, en especial en la zona central.

Tabla 10.5.1-2: Errores. Red 1: (4)-(5)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 1: (4)-(5)-(1)	0,41536	0,05023	0,26046	0,03978

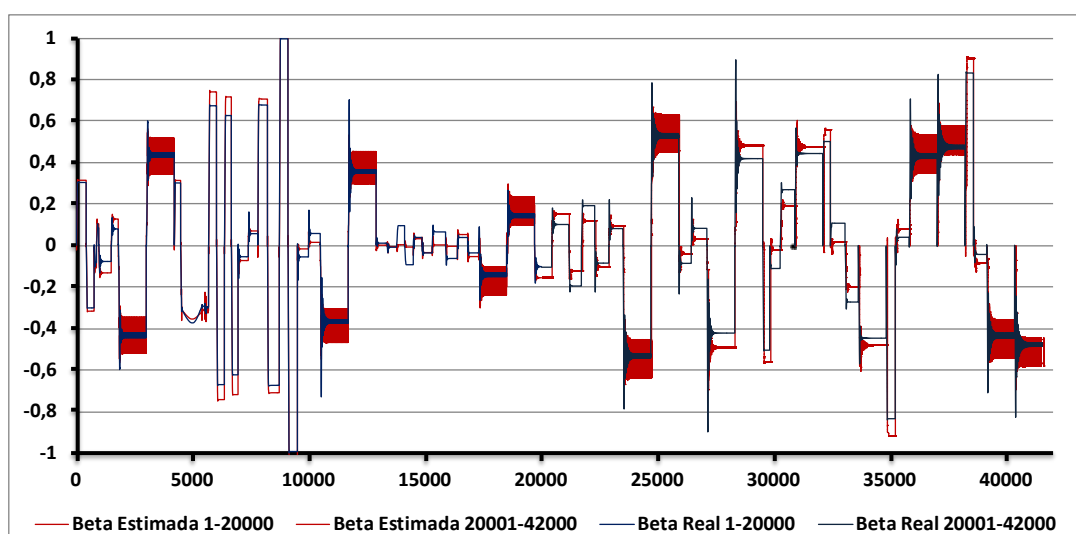


Gráfico 10.5.1-1: Entrenamiento Red 1: (4)-(5)-(1)

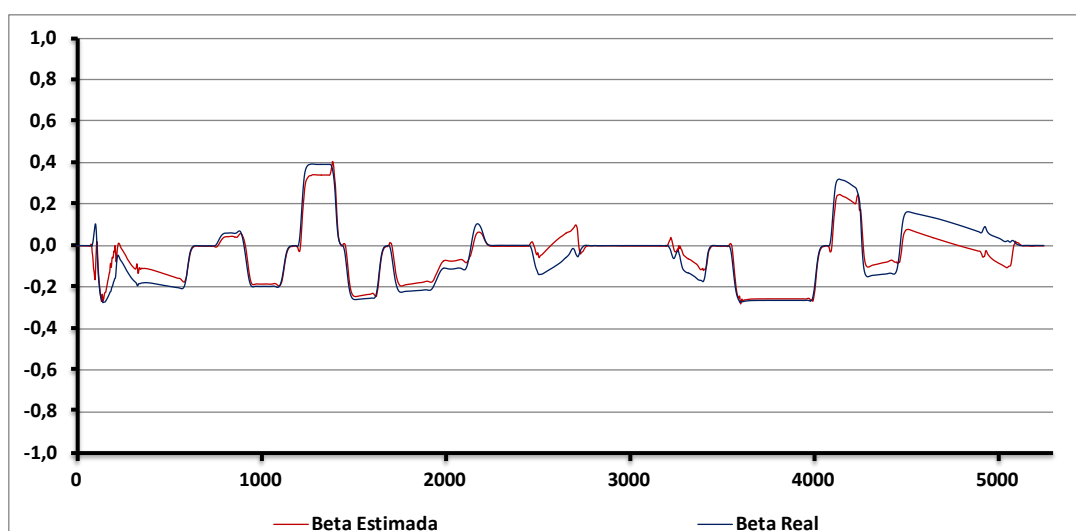


Gráfico 10.5.1-2: Validación Red 1: (4)-(5)-(1)

10.5.2 Red 2: (4)-(10)-(1)

La Red 2 fue entrenada durante casi 170000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0103 registrado en los 10000 ciclos a 0.00207 al final del aprendizaje, datos recogidos en la Tabla 10.5.2-1. Además, se puede apreciar en la Figura 10.5.2-1 cómo ha ido evolucionando el error.

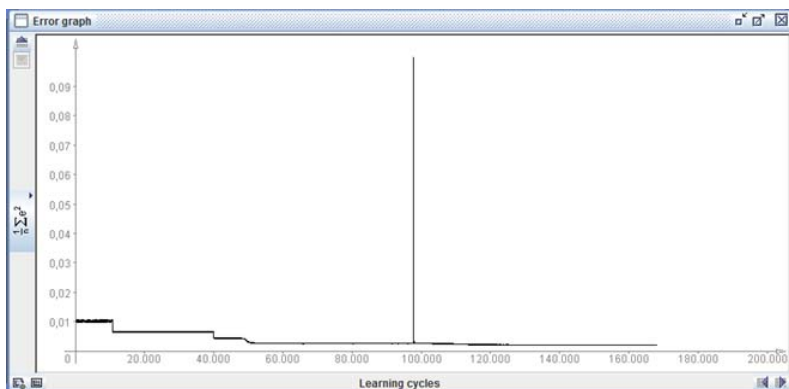


Figura 10.5.2-1: Evolución del MSE de la Red 2: (4)-(10)-(1)

Tabla 10.5.2-1: MSE Red 2: (4)-(10)-(1)

Step	MSE
20000	0.006407375041472755
40000	0.006564997373547152
60000	0.0026729575022586844
80000	0.0025254411030850893
100000	0.0025842933967167524
120000	0.0021416613815444193
130000	0.0020905381905376467
140000	0.002094191611323940
150000	0.002077947728090380
160000	0.002077863457497380

Las estimaciones de la red 2, tanto para el entrenamiento como para la validación, han sido bastante satisfactorias. En cuanto al entrenamiento, cabe destacar que los valores estimados no se ajustan correctamente cuando los valores reales oscilan. Respecto a la validación, los valores estimados coinciden de manera general a los reales.

Tabla 10.5.2-2: Errores. Red 2: (4)-(10)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 2: (4)-(10)-(1)	0,42466	0,04176	0,35422	0,02793

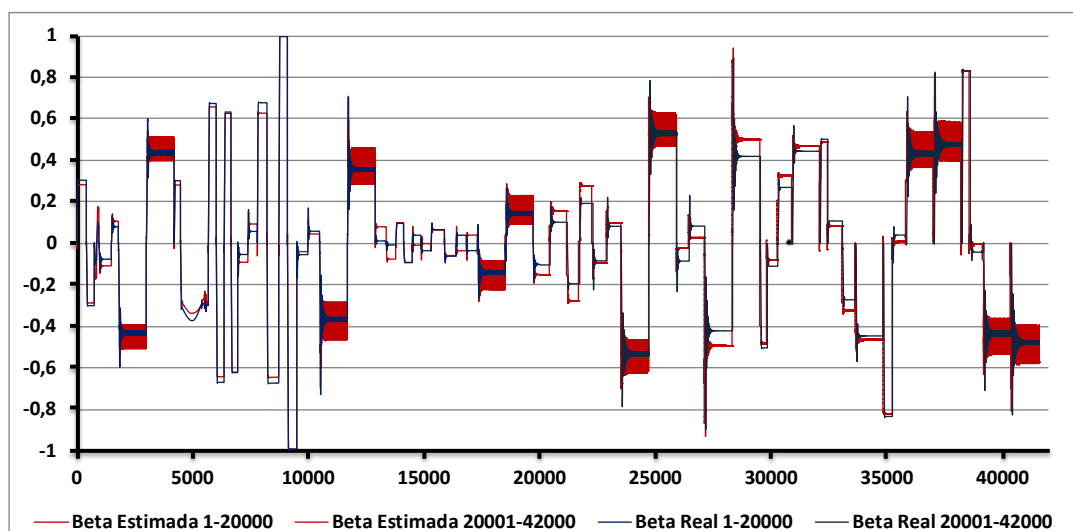


Gráfico 10.5.2-1: Entrenamiento Red 2: (4)-(10)-(1)

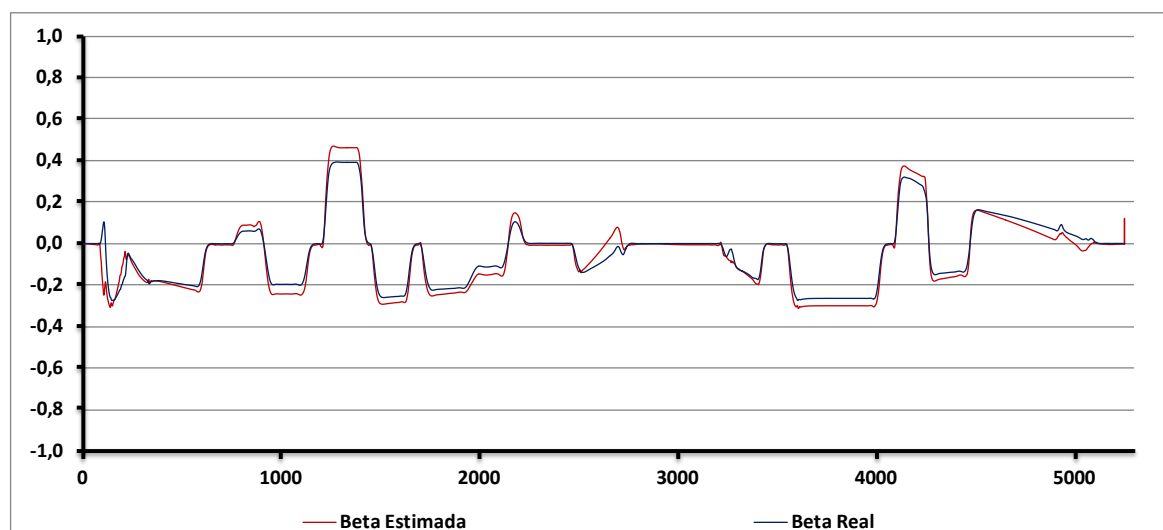


Gráfico 10.5.2-2: Validación Red 2: (4)-(10)-(1)

10.5.3 Red 3: (4)-(15)-(1)

La Red 3 fue entrenada durante 240000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0148 registrado en los 10000 ciclos a 0.0019 al final del aprendizaje, datos recogidos en la Tabla 10.5.3-1. Además, se puede apreciar en la Figura 10.5.3-1 cómo ha ido evolucionando el error.

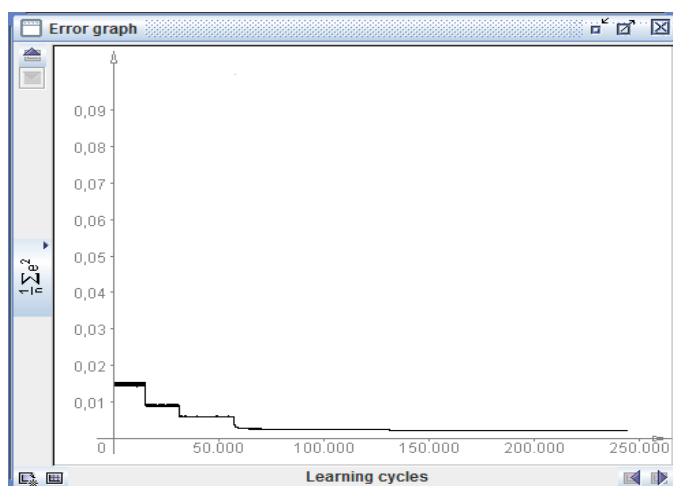


Figura 10.5.3-1: Evolución del MSE de la Red 3: (4)-(15)-(1)

Tabla 10.5.3-1: MSE Red 3: (4)-(15)-(1)

Step	MSE
20000	0.00901674565940193
40000	0.00585182079518187
60000	0.0027495008086141142
80000	0.0024048731318430045
100000	0.0023485306126121066
120000	0.0022895771579455474
140000	0.00212776908373247
160000	0.002096296378349808
180000	0.002107317119388439
200000	0.00196990756120393
220000	0.0019892750865185835
240000	0.001959345586366641

Los valores estimados durante el aprendizaje de la red no se ajustan a los valores reales de manera general.

En cambio, se trata de la red que arroja mejores valores estimados del ángulo de deriva durante la validación. Aunque en los patrones finales se ven algo distorsionados.

Tabla 10.5.3-2: Errores. Red 3: (4)-(15)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 3: (4)-(15)-(1)	0,41979	0,04670	0,39157	0,02725

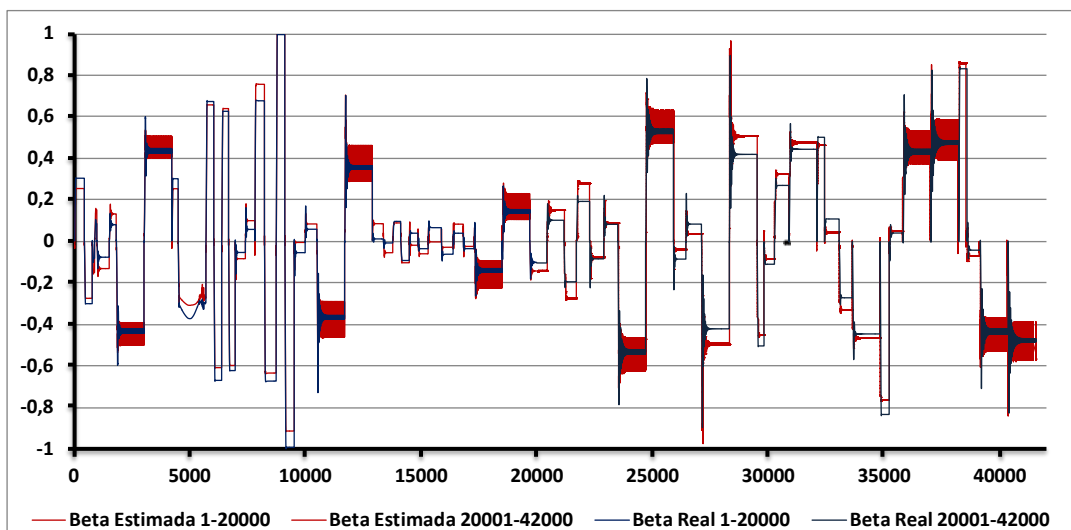


Gráfico 10.5.3-1: Entrenamiento Red 3: (4)-(15)-(1)

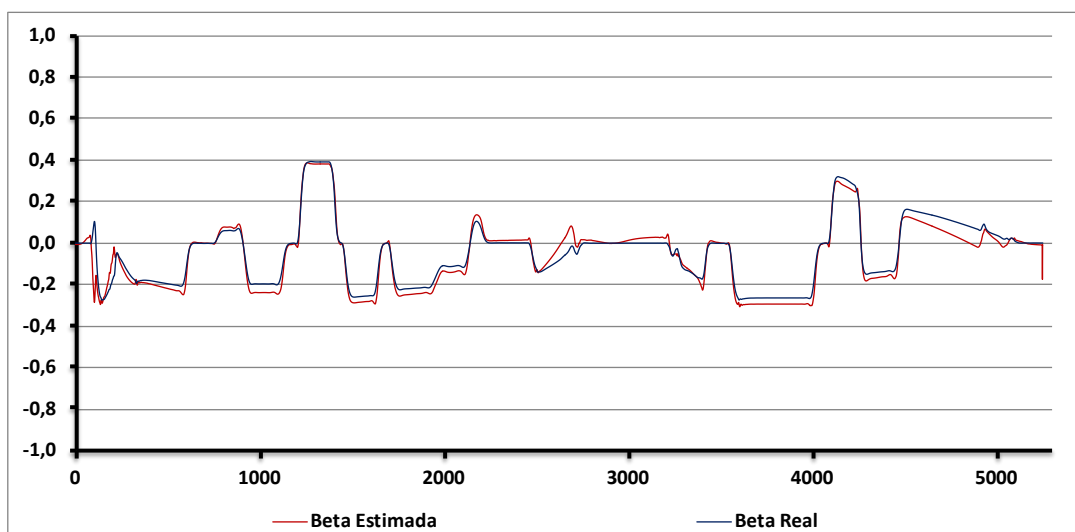


Gráfico 10.5.3-2: Validación Red 3: (4)-(15)-(1)

10.5.4 Red 4: (4)-(20)-(1)

La Red 4 fue entrenada durante 100000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0089 registrado en los 10000 ciclos a 0.0051 al final del aprendizaje, datos recogidos en la Tabla 10.5.4-1. Además, se puede apreciar en la Figura 10.5.4-1 cómo ha ido evolucionando el error.

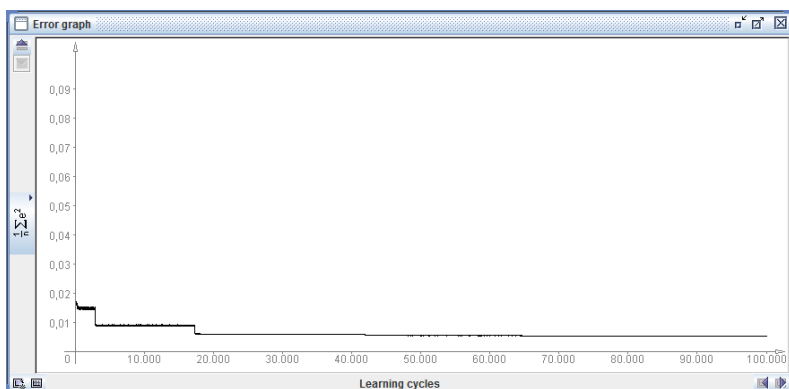


Figura 10.5.4-1: Evolución del MSE de la Red 4: (4)-(20)-(1)

Tabla 10.5.4-1: MSE Red 4: (4)-(20)-(1)

Step	MSE
10000	0.00897836694261547
20000	0.005858433494066607
30000	30.355546951293945
40000	0.005802178514546337
50000	0.005391500309929805
60000	0.0054401654875680305
70000	27.13595199584961
80000	27.191280364990234
90000	0.005213838648827865
100000	0.005195234099289867

A diferencia de la red 3, esta tipología obtiene mejores valores estimados durante el entrenamiento pero, paradójicamente, los valores estimados durante la validación son bastante lejanos respecto a los que deberían de obtenerse.

También cabe destacar que la red ha ajustado los parámetros (pesos y ganancias) de tal forma que para los patrones donde los valores reales del ángulo de deriva oscilan se obtengan valores estimados estabilizados.

Tabla 10.5.4-2: Errores. Red 4: (4)-(20)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 4: (4)-(20)-(1)	0,46523	0,04051	0,27469	0,05686

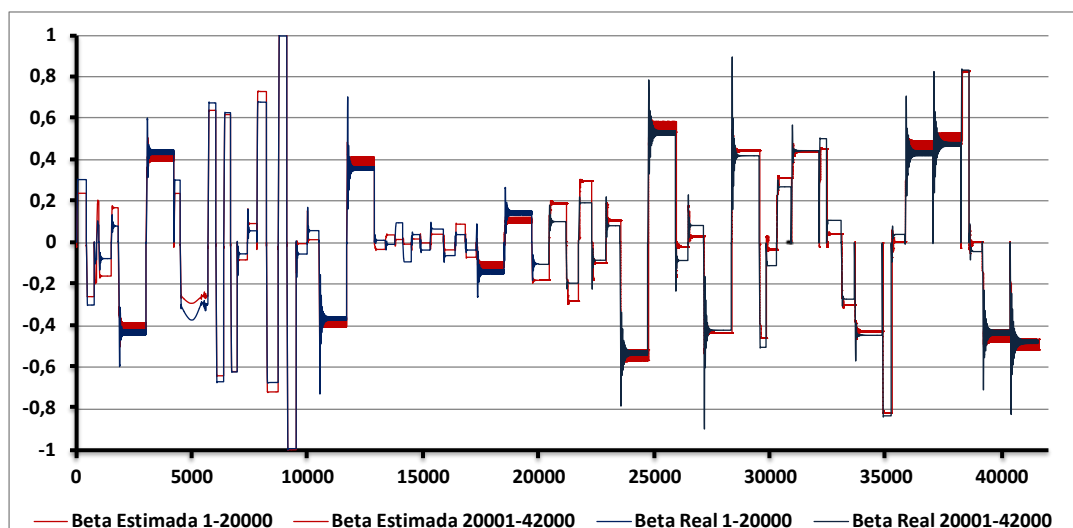


Gráfico 10.5.4-1: Entrenamiento Red 4: (4)-(20)-(1)

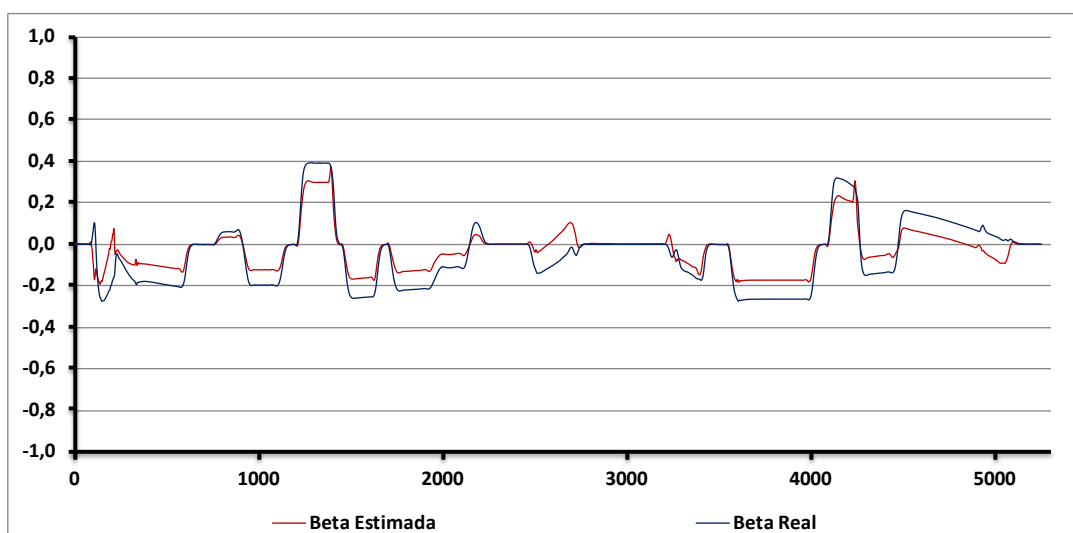


Gráfico 10.5.4-2: Validación Red 4: (4)-(20)-(1)

10.5.5 Red 5: (4)-(25)-(1)

La Red 5 fue entrenada durante 100000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0151 registrado en los 10000 ciclos a 0.0052 al final del aprendizaje, datos recogidos en la Tabla 10.5.5-1. Además, se puede apreciar en la Figura 10.5.5-1 cómo ha ido evolucionando el error.

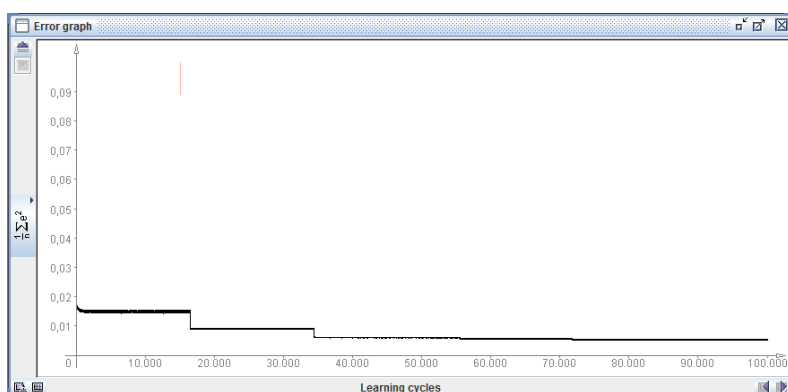


Figura 10.5.5-1: Evolución del MSE de la Red 5: (4)-(25)-(1)

Tabla 10.5.5-1: MSE Red 5: (4)-(25)-(1)

Step	MSE
10000	0.01511576152533446
20000	0.00906767706216301
30000	0.00906786957687933
40000	0.005964740120417911
50000	0.006009376655190678
60000	0.005530111995385365
70000	0.005560204951428523
80000	0.005242820831281393
90000	0.005305851239110147
100000	0.00524298174452541

Los valores estimados por esta red para el entrenamiento como para la validación no terminan de ajustarse adecuadamente.

Tabla 10.5.5-2: Errores. Red 5: (4)-(25)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 5: (4)-(25)-(1)	0,44886	0,04634	0,24758	0,04685

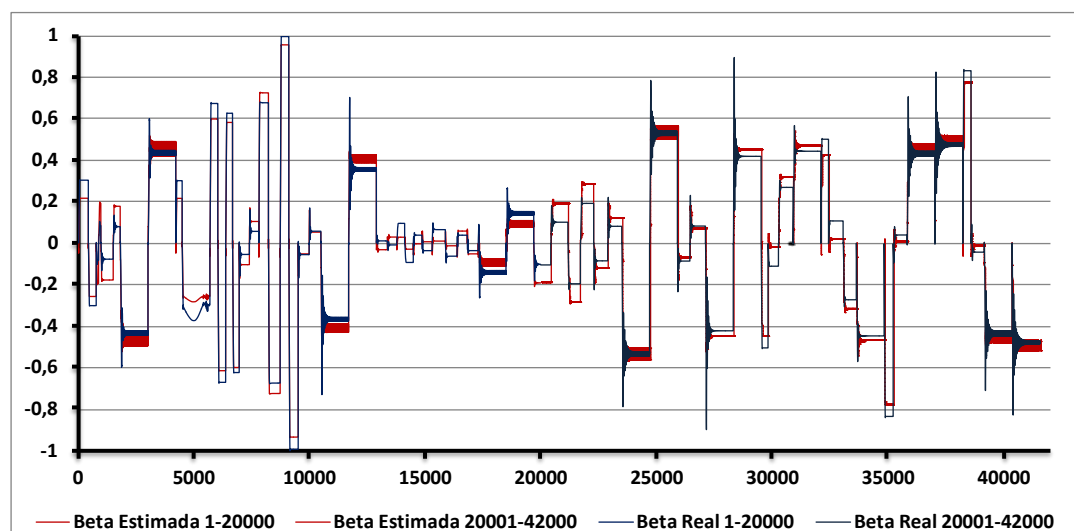


Gráfico 10.5.5-1: Entrenamiento Red 5: (4)-(25)-(1)

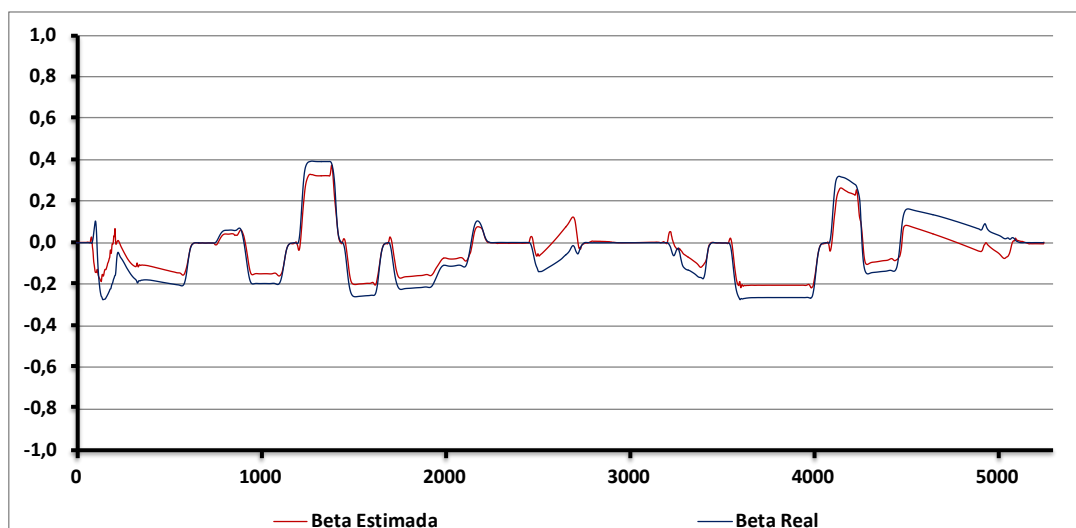


Gráfico 10.5.5-2: Validación Red 5: (4)-(25)-(1)

10.5.6 Red 6: (4)-(5-5)-(1)

La Red 6 fue entrenada durante 170000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0035 registrado en los 20000 ciclos a 0.0021 al final del aprendizaje, datos recogidos en la Tabla 10.5.6-1. Además, se puede apreciar en la Figura 10.5.6-1 como ha ido evolucionando el error.

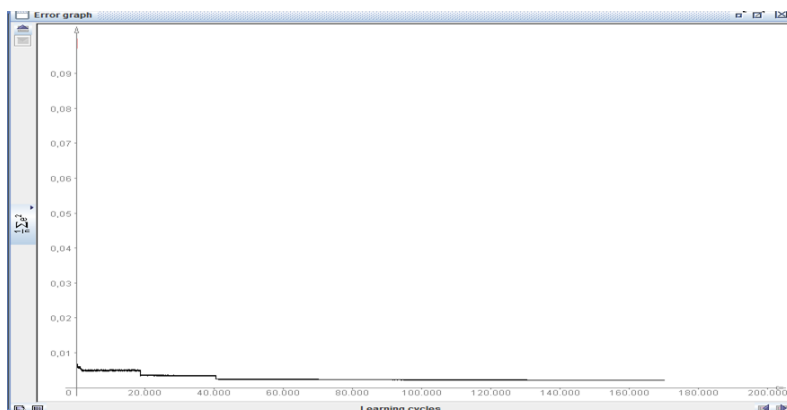


Figura 10.5.6-1: Evolución del MSE de la Red 6: (4)-(5-5)-(1)

Tabla 10.5.6-1: MSE Red 6: (4)-(5-5)-(1)

Step	MSE
20000	0.0035401120727980755
40000	0.003467171152667349
60000	0.0023899160097086234
80000	0.002301582448302258
100000	0.0022392327434515823
120000	0.0021595712435447563
140000	0.002124059179854870
150000	0.002129164452190468
160000	0.002121239747349091
170000	0.0021188436656015483

Se trata de una red con dos capas ocultas con un total de 10 neuronas por lo que cabría esperar que los resultados fueran similares a los obtenidos en la red 2 pero no es así. La red 6, no ha conseguido aprender bien y ajustar los parámetros para determinar el ángulo de deriva de una manera cercana a los valores reales proporcionados. Esta red es la que mayor error medio comete en la validación.

Tabla 10.5.6-2: Errores. Red 6: (4)-(5-5)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 6: (4)-(5-5)-(1)	0,40820	0,04868	0,23882	0,07257

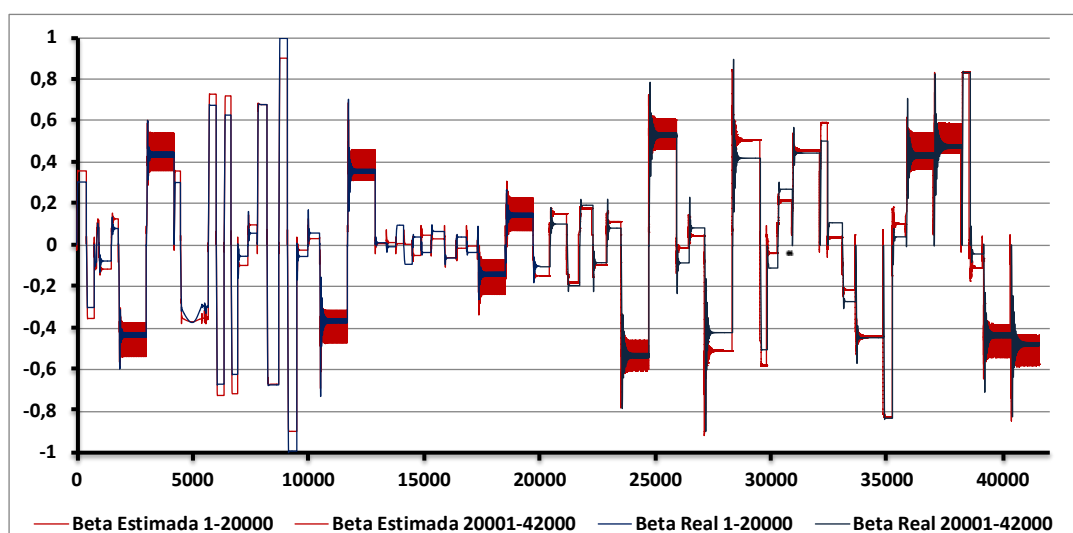


Gráfico 10.5.6-1: Entrenamiento Red 6: (4)-(5-5)-(1)

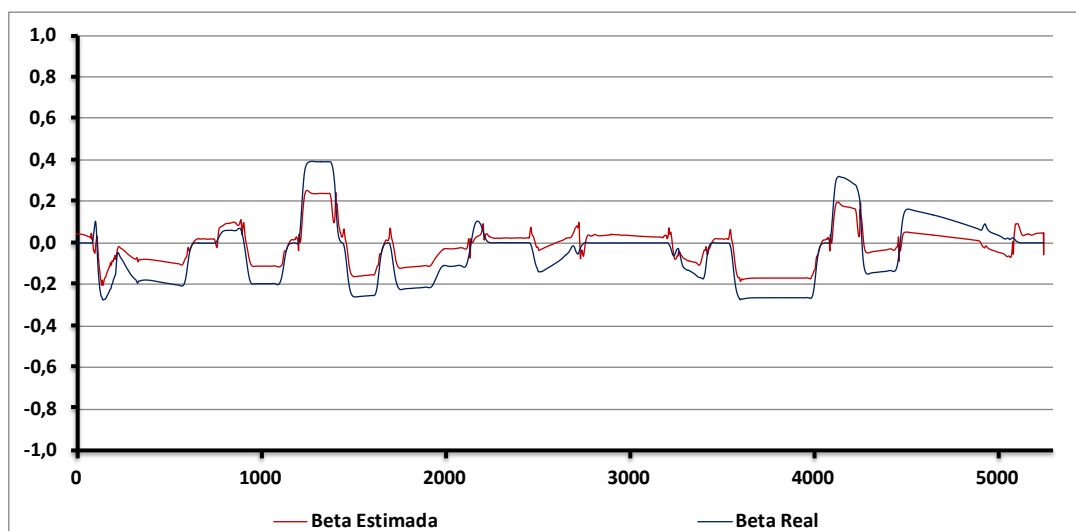


Gráfico 10.5.6-2: Validación Red 6: (4)-(5-5)-(1)

10.5.7 Red 7: (4)-(10-10)-(1)

La Red 7 fue entrenada durante 100000 ciclos consiguiendo disminuir el error cuadrático medio desde 0.0091 registrado en los 10000 ciclos a 0.0028 al final del aprendizaje, datos recogidos en la Tabla 10.5.1-1. Además, se puede apreciar en la Figura 10.5.1-1 como ha ido evolucionando el error.

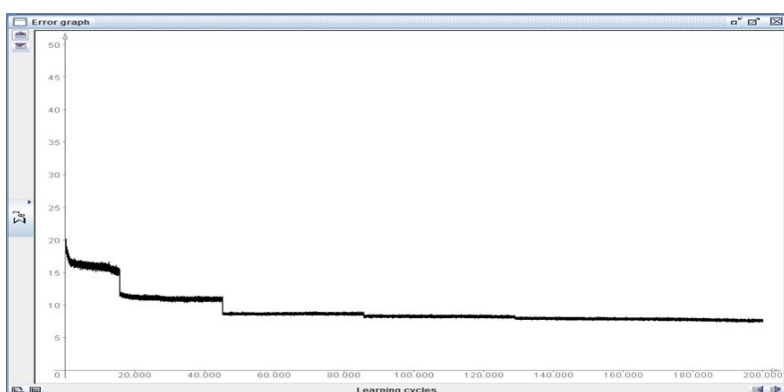


Figura 10.5.7-1: Evolución del MSE de la Red 7: (4)-(10-10)-(1)

Tabla 10.5.7-1: MSE Red 7: (4)-(10-10)-(1)

Step	MSE
20000	0.0021034655549601053
40000	0.002100721310942633
60000	0.0016151922949153659
70000	0.0016672197602358666
80000	0.001649173866610508
100000	0.0015816084658888448
120000	0.001594219632748989
160000	0.0015043497675828674
180000	0.001486137875419098
200000	0.0014527474691971803

Se trata de una red neuronal con dos capas ocultas con un total de 20 neuronas por lo que los resultados obtenidos deberían asemejarse a los de la red 4. En este caso sí es así, incluso los valores estimados se ajustan mejor a los valores reales del ángulo beta.

Tabla 10.5.2-2: Errores. Red 2: (4)-(10)-(1)

TIPOLOGÍA	ENTRENAMIENTO		VALIDACIÓN	
	ERROR MAX	ERROR MEDIO	ERROR MAX	ERROR MEDIO
Red 1: (4)-(10)-(1)	0,39286	0,04015	0,33494	0,05262

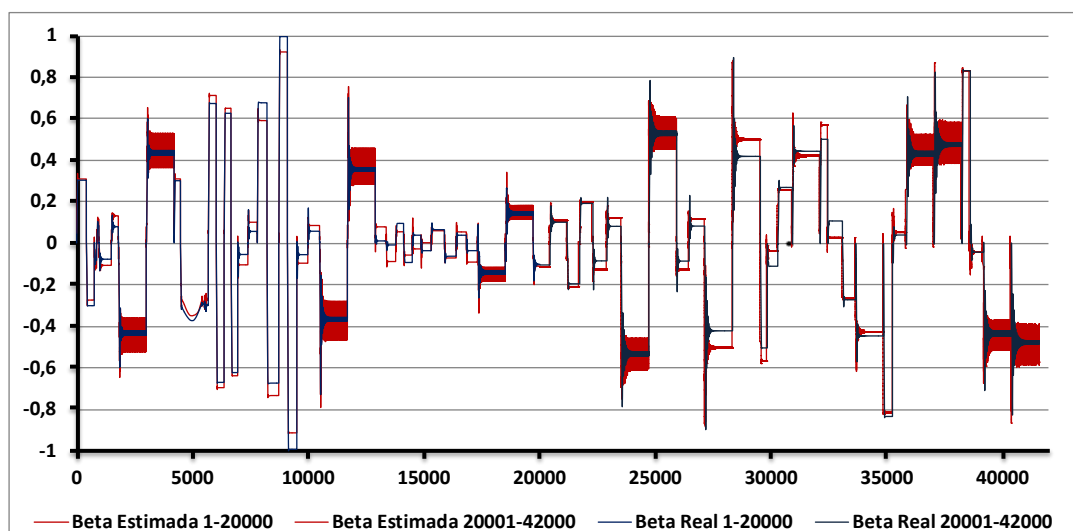


Gráfico 10.5.7-1: Entrenamiento Red 7: (4)-(10-10)-(1)

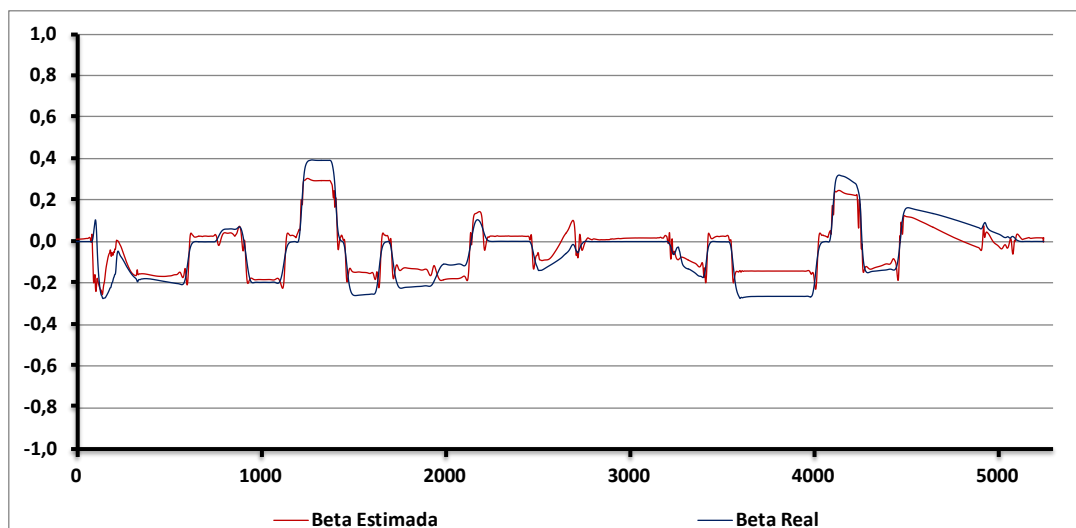


Gráfico 10.5.7-2: Validación Red 7: (4)-(10-10)-(1)

10.6 COMPARATIVA DE LAS REDES

En el apartado 10.5 se presentaron los resultados obtenidos de las redes neuronales entrenadas y validas y en este apartado se hará una comparación de las mismas.

En la tabla 10.6-1 se muestra los errores mínimos MSE registrados por cada una de ellas durante el proceso de aprendizaje.

Tabla 10.6-1: Mínimos MSE para cada red.

TIPOLOGÍA	MSE
Red 1: (4)-(5)-(1)	0,00280
Red 2: (4)-(10)-(1)	0,00208
Red 3: (4)-(15)-(1)	0,00196
Red 4: (4)-(20)-(1)	0,00520
Red 5: (4)-(25)-(1)	0,00524
Red 6: (4)-(5-5)-(1)	0,00212
Red 7: (4)-(10-10)-(1)	0,00145

En la Tabla 10.6-2 se han agrupado los errores máximos y medio para entrenamiento y validación de cada tipología de red propuesta para realizar un análisis comparativo entre las distintas redes. Además, a la derecha de cada columna se ha dispuesto otra, donde se ha introducido el orden en el que está cada red de menor a mayor.

Tabla 10.6-2: Resumen errores.

TIPOLOGÍA	ENTRENAMIENTO				VALIDACIÓN			
	ERROR MAX		ERROR MEDIO		ERROR MAX		ERROR MEDIO	
Red 1: (4)-(5)-(1)	0,41536	3	0,05023	7	0,26046	3	0,03978	3
Red 2: (4)-(10)-(1)	0,42466	5	0,04176	3	0,35422	6	0,02793	2
Red 3: (4)-(15)-(1)	0,41979	4	0,04670	5	0,39157	7	0,02725	1
Red 4: (4)-(20)-(1)	0,46523	7	0,04051	2	0,27469	4	0,05686	6
Red 5: (4)-(25)-(1)	0,44886	6	0,04634	4	0,24758	2	0,04685	4
Red 6: (4)-(5-5)-(1)	0,40820	2	0,04868	6	0,23882	1	0,07257	7
Red 7: (4)-(10-10)-(1)	0,39286	1	0,04015	1	0,33494	5	0,05262	5

Analizando los datos se pueden extraer las siguientes conclusiones preliminares:

- ❖ Los errores máximos durante la fase de entrenamiento son muy semejantes para cualquier tipo de red, alrededor de 0.42, por lo que para decidir cuál de ellas se ajusta mejor a los valores reales del ángulo de deriva es necesario estudiar el error medio.
- ❖ El error medio en el entrenamiento oscila, para las siete tipología definidas, entre 0.04051, y 0.05023. La red que aparentemente mejor ha aprendido y ha sabido ajustar sus pesos y ganancias ha sido la red 7, (4)-(10-10)-(1). No obstante, no es esta red la que genera mejores estimaciones con los patrones de validación. La red 1 es la que peor ha aprendido.
- ❖ Los errores máximos obtenidos para los patrones de validación están comprendidos entre 0.23882 y 0.39157, bastante inferiores a los recogidos durante el entrenamiento.
- ❖ El error medio en las validaciones están entre 0.02725 y 0.07257, inferiores a los tomados en el entrenamiento. La tipología que menor error ha obtenido es la red 3, en cambio, se trata de la segunda red que mayor error medio ha obtenido durante su entrenamiento. La red que mayor error medio ha obtenido en las validaciones es la red 6 cuyo valor es muy superior en comparación al resto de redes.
- ❖ Las redes 6 y 7 fueron propuestas para confirmar la teoría de que es indiferente cuantas capas ocultas haya siempre que tengan la misma cantidad de neuronas. La red 6 corresponde a la red 2, ambas con 10 neuronas en las capas ocultas, y la red 7 corresponde a la red 5, ambas con 20 neuronas en las capas ocultas. Tras analizar los resultados obtenidos no es concluyente ya que la primera pareja no ofrece resultados similares, mientras que la segunda sí.

- ❖ De entre todas las tipologías, la red 2 y la red 3 son las que mejores estimaciones han ofrecido en la fase de validación. Aunque la red 3 ha obtenido menor error medio, es la red 2 la que tiene menor error medio en entrenamiento y en validación en conjunto. Además, la red 2 (15 neuronas en total) es más simple que la red 3 (20 neuronas en total), además, ha necesitado menos ciclos para aprender, 170000 frente a 240000.

Por tanto, la red elegida como apta para la estimación del ángulo de deriva de un vehículo es la red 2 cuya estructura es (4)-(10)-(1). En la Figura 10.6-1 se muestra los parámetros definidos por la red durante el entrenamiento.

SNNS network definition file V1.4-3D
generated at Tue Apr 09 19:46:22 2013

network name : red-2 entrenada
source files :
no. of units : 15
no. of connections : 50
no. of unit types : 0
no. of site types : 0

learning function : Std_Backpropagation
update function : Topological_Order

unit default section :

act	bias	st	subnet	layer	act func	out func
0.00000	0.00000	h	0	1	Act_Logistic	Out_Identity

unit definition section :

no.	typeName	unitName	act	bias	st	position	act func	out func	sites
1		noName	0.47594	0.06694	i	1, 1, 1	Act_TanH		
2		noName	0.66803	-0.09298	i	1, 2, 1	Act_TanH		
3		noName	0.70528	0.00340	i	1, 3, 1	Act_TanH		
4		noName	0.70374	0.03260	i	1, 4, 1	Act_TanH		
5		noName	-0.57150	-0.02237	h	2, 1, 1	Act_TanH		
6		noName	-0.00000	-0.00000	h	2, 2, 1	Act_TanH		
7		noName	-0.85800	0.18507	h	2, 3, 1	Act_TanH		
8		noName	-0.97819	0.15927	h	2, 4, 1	Act_TanH		
9		noName	0.65958	-0.22065	h	2, 5, 1	Act_TanH		
10		noName	-0.21183	0.02786	h	2, 6, 1	Act_TanH		
11		noName	-0.00000	-0.00000	h	2, 7, 1	Act_TanH		
12		noName	-1.00000	-8.36737	h	2, 8, 1	Act_TanH		
13		noName	-0.00000	-0.00000	h	2, 9, 1	Act_TanH		
14		noName	0.99649	8.28443	h	2,10, 1	Act_TanH		
15		noName	0.12014	-27.74547	o	3, 1, 1	Act_Identity		

connection definition section :

target	site	source:weight
5		1: 0.02041, 2: 1.65959, 3:-2.68300, 4: 0.20820
6		1: 0.00000, 2: 0.00000, 3: 0.00000, 4:-0.00000
7		1:-2.57880, 2:-0.44547, 3: 0.12390, 4:-0.04723
8		1:-0.15225, 2:-5.90925, 3: 2.91775, 4:-0.64076
9		1: 2.61058, 2:-0.46385, 3: 0.14133, 4:-0.02779
10		1:-0.02339, 2:-1.36513, 3: 2.05145, 4:-1.08947
11		1: 0.00000, 2:-0.00000, 3: 0.00000, 4: 0.00000
12		1: 0.94829, 2: 0.02811, 3:-2.43766, 4:-4.27800
13		1: 0.00000, 2:-0.00000, 3: 0.00000, 4: 0.00000
14		1:-0.92412, 2: 0.03612, 3:-2.40821, 4:-4.26048
15		5: 0.56958, 6: 0.00000, 7:-2.35514, 8:-0.18421, 9:-2.27045, 10: 0.78554, 11: 0.00000, 12: 2.57120, 13: 0.00000, 14: 2.48863

layer definition section :

layer	unitNo.
2	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
3	15

Figura 10.6-1: Parámetros de la Red 2: (4)-(10)-(1)



11 PRESUPUESTO



11.1 INTRODUCCIÓN

En este capítulo se aborda el aspecto económico. Se realiza un desglose del presupuesto en el que se tienen en cuenta los gastos de personal y de recursos informáticos.

11.2 PRESUPUESTO DETALLADO

11.2.1 PERSONAL

NOMBRES	CATEGORÍA	HORAS	COSTE UNITARIO	COSTE TOTAL
Guillermo Maudes Comendador	Ingeniero Junior	60 h	18 €/h	1080 €
María Jesús López Boada	Ingeniero Senior	10 h	40 €/h	400 €
Beatriz López Boada	Ingeniero Senior	10 h	40 €/h	400 €
				1880 €

11.2.2 LICENCIAS Y EQUIPOS DE INFORMÁTICOS

UNIDADES	DESCRIPCIÓN	COSTE TOTAL	AMORTIZACIÓN RELATIVA IMPUTABLE	COSTE AMORTIZACIÓN IMPUTABLE
3	Equipo informático	1000 €	10%	300 €
3	Licencia Microsoft Office	180 €	10%	54 €
1	Licencia CarSim	15000 €	5%	1500 €
1	Licencia Java-NNS	0 €	10%	0 €
				1854 €

11.3 PRESUPUESTO GLOBAL

CONCEPTO	COSTE
Personal	1880 €
Licencias y Equipos Informáticos	1854 €
SUBTOTAL	3734 €
IVA (21%)	784.14 €
TOTAL	4518.14 €

12 CONCLUSIONES Y TRABAJOS FUTUROS



12.1 CONCLUSIONES

En el presente Proyecto Fin de Carrera se han entrenado hasta siete redes neuronales artificiales con la finalidad de encontrar un modelo no paramétrico que estimase el ángulo de deriva del un vehículo en función de la velocidad longitudinal, la aceleración lateral, la velocidad de guiñada y el ángulo de dirección. De este estudio se pueden obtener las siguientes conclusiones:

- ❖ Se ha llevado a cabo un acercamiento a los sistemas de seguridad instalados en los vehículos centrando mayor atención en los sistemas de control de estabilidad. Se ha conocido la problemática existente a la hora de obtener algunos parámetros clave en la dinámica de los automóviles para mejorar los sistemas actuales y desarrollar nuevos que permitan reducir la siniestralidad en las carreteras.
- ❖ Se ha realizado un acercamiento al ángulo de deriva de un vehículo tratando de entender qué es, cómo se forma, de qué variables depende y que estrategias son seguidas para su medición y estimación.
- ❖ Se ha estudiado la posibilidad de estimar el ángulo de deriva de un vehículo a través de la técnica de redes neuronales artificiales. Los datos obtenidos, introduciendo las siguientes variables de entrada: Velocidad longitudinal, Aceleración lateral, Velocidad de guiñada del vehículo y el Ángulo de dirección del vehículo, reflejan de manera positiva su validez, tanto para situaciones en las que el vehículo tiene un comportamiento lineal como no lineal.
- ❖ Se deduce del análisis realizado a los datos obtenidos que una red puede haber aprendido muy bien para los valores de los patrones de entrenamiento, consiguiendo minimizar el MSE, pero no ser capaz de estimar el ángulo de deriva con otro set de datos de entrada, como han podido ser los patrones de validación. Puede decirse que hay que llegar a un compromiso durante el aprendizaje ya que las redes que han sabido ajustar sus parámetros para obtener los patrones de entrenamiento de manera más precisa se vuelven demasiado rígidas, no permitiendo extrapolar sus conocimientos a nuevos sets de datos de entrada. Esto es lo que ha ocurrido con la red 4, cuya estructura es (4)-(20)-(1), y con la red 7, cuya estructura es (4)-(10-10)-(1). Estas redes han obtenido buenos resultados en la etapa de entrenamiento pero en cambio no arrojan buenos resultados durante la validación. En cambio, la red 3, cuya estructura es (4)-(15)-(1), no ha sido la que mejores resultados ha obtenido durante el aprendizaje pero es la que mejores valores ofrece con nuevos set de datos de entrada.

- ❖ Se ha seleccionado la red 2 como apta para la estimación del ángulo de deriva de entre todas las redes propuestas. Se trata de una estructura bastante simple, (4)-(10)-(1), debido a que está formada por cuatro neuronas en la capa de entrada, diez neuronas en la capa oculta y una neurona en la capa de salida. Esta red ha ofrecido resultados del ángulo de deriva muy próximos a los reales. En la Tabla 12.1-1 se recogen los valores normalizados para el error máximo y error medio obtenidos en la validación de la red y los valores reales del ángulo de deriva al desnormalizar los anteriores. Los valores reales han sido calculados al multiplicar los primeros por el valor con el que se normalizó la variable β , 3.898210° , ver Tabla 10.3-1.

Tabla 12.1-1: Errores normalizados y reales obtenidos para la red 2

	NORM.	REAL
ERROR MÁX.	0,35422°	1,38°
ERROR MEDIO	0,02793°	0,11°

- ❖ Es necesario continuar con el estudio iniciado en este documento para comprobar mediante ensayos reales que este tipo de sistema es capaz de negociar y estimar el ángulo de deriva de un vehículo cumpliendo con los requerimientos exigidos por los sistemas de seguridad en los automóviles.
- ❖ No se puede corroborar, por falta de más análisis, que dos redes con diferente número de capas ocultas pero con la misma cantidad de neuronas totales arrojen idénticos resultados. Esto ha sido analizado a través de las redes 2 y 6 y de las redes 4 y 7.

12.2 TRABAJOS FUTUROS

A partir de la realización y de las conclusiones de este Proyecto Fin de Carrera han surgido algunas ideas que pueden dar lugar a futuros trabajos:

- ❖ Realizar más simulaciones en CarSim variando otros parámetros que no se han considerado en este estudio para que la red neuronal tenga mayor conocimiento sobre todas las condiciones en las que puede trabajar un vehículo. Se pueden considerar en futuros estudios las siguientes maniobras y parámetros:
 - Simular otro tipo de maniobras como por ejemplo cambio simple de carril, doble cambio de carril, rotonda, circuitos, slalom...
 - Realizar las maniobras suponiendo que la carretera no es horizontal sino que tiene pendientes, tanto de peralte como de inclinación.
 - El vehículo puede verse modificado por lo que es de interés que la red neuronal conozca cómo se comporta al variar parámetros como: el tipo de neumáticos, la suspensión, la carga, parámetros dimensionales...
 - Variar los parámetros de la simulación: no considerando una velocidad longitudinal sino que se acelera y se frena, el coeficiente de adherencia varía...
- ❖ Obtener los patrones de entrenamiento y validación mediante ensayos reales de maniobras con un vehículo completamente equipado con sensores a partir de tomados a partir de maniobras reales.
- ❖ Se pueden considerar mayor número de parámetros de entrada a parte de los cuatro escogidos en este proyecto fin de carrera para que las redes neuronales tenga mayor información acerca de la dinámica del vehículo, del entorno por el que se desplaza y de la interacción entre el vehículo y el entorno.
- ❖ Generar otro tipo de tipología de redes variando el número de capas ocultas y de neuronas que hay en ellas, el modo en el que están conectadas...
- ❖ Realizar otros entrenamientos de redes neuronales variando las funciones de entrada, activación y salida de las neuronas, así como, el método empleado para el aprendizaje.
- ❖ Utilización de otros programas de redes neuronales que hay en el mercado.
- ❖ Implantar el sistema entrenado y validado en un vehículo real y comprobar si es capaz de gestionar los datos obtenidos a través de los sensores para obtener el ángulo de deriva del vehículo con la rapidez precisión y exactitud requerida.



13 BIBLIOGRAFÍA



- [1] A. Hac and M. Simpson: *Estimation of vehicle sideslip angle and yaw rate*.(2000) presented at the SAE World Congr., Detroit, MI, SAE Paper No. 2000-01-0696.
- [2] A. Loría, E. Panteley, D. Popović, and A. R. Teel: *A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems*. (2005) IEEE Trans. Automat. Contr., vol. 50, no. 2, pp. 183 –198
- [3] A. Moreno y J. Alonso: *Aprendizaje automático* (1994). Universidad Politécnica de Catalunya.
- [4] A. Nishio, K. Tozu, H. Yamaguchi, K. Asano and Y. Amano: *Development of vehicle stability control system based on vehicle side slip angle estimation*.(2001) SAE Paper No. 2001-01-0137
- [5] A. Suissa, Z. Zomotor, and F. Böttiger: *Method for determining variables characterizing vehicle handling*. (1996) U.S. Patent 5 557 520.
- [6] A. T. Van Zanten et al.: *Control aspects of the Bosch VDC*. (1996) In Proc.AVEC, pp. 573–607.
- [7] A. T. Van Zanten: *Bosch ESP systems: 5years of experience*. (2000) SAE Paper No. 2000-01-1633.
- [8] A. von Vietinghoff, M. Hiemer, and U. Kiencke: *Nonlinear observer design for lateral vehicle dynamics*. (2005) In Proc. IFAC World Congr., Prague, Czech Republic, pp. 988 –993.
- [9] A. von Vietinghoff, S. Olbrich, and U. Kiencke: *Extended Kalman filter for vehicle dynamics determination based on a nonlinear model combining longitudinal and lateral dynamics*. (2007) In Proc. SAE World Congr., Detroit, MI, Paper 2007-01-0834.
- [10] A. Y. Ungoren, H. Peng, and H. E. Tseng, “A study on lateral speed estimation methods,” Int. J. Vehicle Autonom. Syst., vol. 2, nos. 1–2, pp. 126 –144, 2004.
- [11] Auto10: *Los mejores sistemas de seguridad para tu coche*. (Última consulta: en enero 2013).<http://www.auto10.com/tecnologia-coches/los-mejores-sistemas-de-seguridad-para-tu-coche>
- [12] Autocasión: *Cómo funciona el ESP*. (Última consulta: enero 2013) <http://www.autocasion.com/actualidad/reportajes/98912/como-funciona-el-esp/>
- [13] Autofacil: *Qué ha aportado Bosch por la seguridad... y qué aportará*. (Última consulta: enero 2013) <http://www.autofacil.es/seguridad/2012/09/25/aportado-bosch-seguridad-aportara/10922.html>
- [14] B. C. Chen and F. C.Hsieh. *Sideslip angle estimation using extended Kalman filter. Vehicle System Dynamics*. (2008). 46, Supplement, 353–364.
- [15] Binh Minh Nguyen, Kanghyun Nam, Hiroshi Fujimoto, Yoichi Hori: *Proposal of cornering stiffness estimation without vehicle side slip angle using lateral force sensor*. Universidad de Tokio

- [16] C. Sentouh, Y. Sebsadji, S. Mammar, and S. Glaser: *Road bank angle and faults estimation using unknown input proportional - integral observer*. (2007) In Proc. European Control Conf., Kos, Greece, pp. 5131 –5138.
 - [17] D. M. Bevly R. Sheridan and J. C. Gerdes: *Integrating INS Sensors with GPS Velocity Measurements for Continuous Estimation of Vehicle Sideslip and Tire Cornering Stiffness*. (2001). Proceedings of the American Control Conference, Arlington, VA.
 - [18] D. M. Bevly, et. al.: *The Use of GPS Based Velocity Measurements for Improved Vehicle State Estimation*. (2000). Proceedings of the 2000 ACC, Chicago, IL.
 - [19] D. M. Bevly, J. C. Gerdes, and C. Wilson: *The use of GPS based velocity measurements for measurement of sideslip and wheel slip*. (2002). Vehicle Syst. Dyn., vol. 38, no. 2, pp. 127 –147.
 - [20] D. M. Bevly: *Global positioning system (GPS): A low cost velocity sensor for correcting inertial sensor errors on ground vehicles*. (2004) J. Dyn. Syst. Meas. Control, vol. 126, no. 2, pp. 255 –264.
 - [21] D. Piyabongkarn, R. Rajamani, J. A. Grogg and J. Y. Lew: *Development and experimental evaluation of a slip angle estimator for vehicle stability control*. (2006). Proc. PACC, 366–5371.
 - [22] D.W. Pi, N.Chen, J. X. Xang y B. J. Zhang: *Design and evaluation of sideslip angle observer for vehicle stability control*. (2010)
 - [23] David Juberías Fernández: *Diseño, Análisis Dinámico Y Modelado Del Tren Delantero De Un Car Cross*. (2011). Universidad Carlos III de Madrid.
 - [24] David M. Bevly, Jihan Ryu y Christian Gerdes: *Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness*. (2006)
 - [25] Dirección General de Tráfico (DGT): *Las principales cifras de la siniestralidad vial en España 2011*. Observatorio Nacional de Seguridad Vial. España.
 - [26] E. Castillo y A. Cobo: *Introducción a las redes funcionales con aplicaciones: un nuevo paradigma neuronal*. (1999)
 - [27] *Evolution of electronic control systems for improving the vehicle dynamic behavior*. (2002) In Proc. AVEC, pp. 7–15.
 - [28] F. Cheli, E. Sabbioni, M. Pesce and S. Melzi: *A methodology for vehicle sideslip angle identification: Comparison with experimental data*. Vehicle System Dynamics. (2007). 45, 6, 549–563.
 - [29] Felipe Borja Muñoz Águila: *Modelo directo no paramétrico de amortiguador magneto-reológico basado en redes neuronales*. (2012). Universidad Carlos III de Madrid.
 - [30] Fernando Izaurieta y Carlos Saavedra: *Redes neuronales artificiales*. Universidad de Concepción, Chile.
-

- [31] Francisco Aparicio Izquierdo, Carlos Vera Alvarez, Vicente Díaz López: *Teoría de los vehículos automóviles*. (1995) Universidad Carlos III de Madrid
- [32] G. Baffet, A. Charara and D. Lechner: *Experimental evaluation of a sliding mode observer for tire road forces and an extended kalman filter for vehicle sideslip angle*. (2007). Proc. 46th IEEE Conf. Decision and Control, 3877–3882.
- [33] Guillaume Baffet, Ali Charara y Daniel Lechner: *Estimation of vehicle sideslip, tire force and wheel cornering stiffness*. (2009)
- [34] Guillermo Maudes Comendador: *Influencia de la defensa de un todoterreno en las lesiones sufridas por un peatón en caso de atropello*. (2010). Universidad Carlos III de Madrid
- [35] H. E. Tseng: *Dynamic estimation of road bank angle*. (2001) Vehicle Syst. Dyn., vol. 36, no. 4, pp. 307 –328.
- [36] H. H. Kim y J. Ryu: *Sideslip angle estimation considering short-duration longitudinal velocity variation*. (2011)
- [37] H. Lee: *Reliability indexed sensor fusion and its application to vehicle velocity estimation*. (2006). J. Dyn. Syst. Meas. Control, vol. 128, no. 2, pp. 236 –243.
- [38] H. Sasaki and T. Nishimaki: *A side-slip angle estimation using neural network for a wheeled vehicle*. (2000). SAE Papers 2000-01-0695, Detroit, Michigan.
- [39] Haiping Du y Nong Zhang. *Robust vehicle stability control based on sideslip angle estimation*. (2007)
- [40] Håvard Fjær, Lars Imsland, Tor A. Johansen, Jens C. Kalkkuhl y Avshalom Suissa: *Vehicle sideslip estimation*. (2009)
- [41] Håvard Fjær, Lars Imsland, Tor A. Johansen, Thor I. Fossen, Jens C. Kalkkuhl y Avshalom Suissa: *Nonlinear vehicle side-slip estimation with friction adaptation*. (2007)
- [42] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. (2008) New York : McGraw Hill.
- [43] J. Farrelly and P. Wellstead: *Estimation of vehicle lateral velocity*. (1996). In Proc. IEEE Int. Conf. Control Application, Dearborn, MI, 552–557.
- [44] J. Fernández Fernández: *Las Redes Neuronales Artificiales*. (2008)
- [45] J. Ryu and J. C. Gerdes: *Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control*. (2004) J. Dyn. Syst. Meas. Control, vol. 126, no. 2, pp. 243 –254.
- [46] J. Stephant, A. Charara and D. Meizel *Virtual sensor: Application to vehicle sideslip angle and transversal forces*. (2004). IEEE Trans. Industrial Electronics 51, 2, 278–289.

- [47] J. Zuurbier and P. Bremmer: *State estimation for integrated vehicle dynamics control*, 6th International Symposium on Advanced Vehicle Control. (2000). Ann Arbor, Michigan, pp. 371–375.
- [48] Jasvipul Singh Chawla: *Estimation of side slip in vehicles*. (2006). Instituto de tecnología de India, Bombay.
- [49] Jihan Ryu, Eric J. Rossetter y J. Christian Gerdes: *Vehicle Sideslip and Roll Parameter Estimation using GPS*. (2002) Universidad de Stanford.
- [50] K. Park, S. J. Heo and I. Baek: *Controller design for improving lateral vehicle dynamic stability*. (2001). JSAE Review, 22, 481–486.
- [51] L. Imsland, H. F. Grip, T. A. Johansen, T. I. Fossen, J. C. Kalkkuhl, and A. Suissa: *Nonlinear observer for vehicle velocity with friction and road bank angle adaptation—Validation and comparison with an extended Kalman filter*. (2007) In Proc. SAE World Congr., Detroit, MI, Paper 2007-01-0808.
- [52] L. Imsland, T. A. Johansen, H. F. Grip, and T. I. Fossen: *On nonlinear unknown input observers—Applied to lateral vehicle velocity estimation on banked roads*. (2007) Int. J. Control, vol. 80, no. 11, pp. 1741–1750.
- [53] L. Imsland, T. A. Johansen, T. I. Fossen, H. F. Grip, J. C. Kalkkuhl, and A. Suissa, “Vehicle velocity estimation using nonlinear observers,” Automatica, vol. 42, no. 12, pp. 2091–2103, 2006.
- [54] L. R. Ray: *Nonlinear tire force estimation and road friction identification: Simulation and experiments*. (1997). Automatica, vol. 33, no. 10, pp. 1819–1833.
- [55] M. C. Best, T. J. Gordon, and P. J. Dixon: *An extended adaptive Kalman filter for real-time state estimation of vehicle handling dynamics*. (2000). Vehicle Syst. Dyn., vol. 34, no. 1, pp. 57–75.
- [56] M. Hiemer, A. von Vietinghoff, U. Kiencke, and T. Matsunaga: *Determination of the vehicle body slip angle with non-linear observer strategies*. (2005). In Proc. SAE World Congr., Detroit, MI, Paper 2005-01-0400.
- [57] M. Milanese, D. Regruto and A. Fortina: *Direct virtual sensor (DVS) design in vehicle sideslip angle estimation*. (2007). Proc. American Control Conf., 3654–3658
- [58] M.C. Best: *An extended adaptive Kalman filter for real time state estimation of vehicle handling dynamics*. (2000). Veh. Syst. Dyn. 34(1), pp. 57–75.
- [59] Mario Milanese, Diego Regruto, Andrea Fortina: *Direct virtual sensor (DVS) design in vehicle sideslip angle estimation*. (2007)
- [60] Mecánicos Costarica. *ESP*. (Última consulta: enero 2013)
<http://www.mecanicoscostarica.com/ARTICULOS/ESP.htm>

- [61] Moustapha Doumiati, Alessandr Correa Victorino, ali Charara, Member, IEEE y Daniel Lechner: *Onboard real-time estimation of vehicle lateral tire-road forces and sideslip angle*. (2010)
- [62] OMS: <http://www.who.int/es/> (Última consulta: enero 2013)
- [63] P. J. TH. Venhovens and K. Naab: *Vehicle dynamics estimation using Kalman filters*. (1999). Vehicle Syst. Dyn., vol. 32, no. 2, pp. 171 –184.
- [64] R. Rojas: *Neural Network* 1996
- [65] Rafael Ibáñez de Aldecoa García: *Estudio y simulación con Simulink de vuelco de un vehículo pesado en adelantamiento*. Universidad Carlos III de Madrid.
- [66] Rubén Valverde García: *Estudio técnico de un vehículo de producción para su uso en competición en el campeonato de España de Gran Turismo*. (2012) Universidad Carlos III de Madrid.
- [67] Rusty Anderson y David M. Bevlly: *Estimation of slip angles using a model based estimator and GPS*. (2004).
- [68] S. Melzi y E.Sabbioni. *Mechanical systems and signal processing*. (2010)
- [69] Seguridad Vial: http://es.wikipedia.org/wiki/Seguridad_vial Última consulta: enero 2013)
- [70] Seung-Han You, Jin-Oh Hahn y Hyeongcheol Lee: *Control engineering practice*. (2009)
- [71] Sistema seguridad: http://www.seguridad-vial.net/sistemas_seguridad.asp?sist=1 (Última consulta: enero 2013)
- [72] Stuttgart Neural Network Simulator. *Java-NNS*. (Última consulta: enero 2013) <http://www.ra.cs.uni-tuebingen.de/SNNS/>
- [73] T. A. Wenzel, K.J. Burnham, M. V. Blundell and R. A. Williams: *Dual extended kalman filter for vehicle state and parameter estimation*. (2006). Vehicle System Dynamics 44, 2, 153–171.
- [74] T. Hiraoka, H. Kumamoto and O. Nishihara: *Sideslip angle estimation and active front steering system based on lateral acceleration data at centers of percussion with respect to front/rear wheels*. (2004). JSAE Review, 25, 37–42.
- [75] U. Kiencke and A. Daiß: *Observation of lateral vehicle dynamics*. (1997) Control Eng. Pract., vol. 5, no. 8, pp. 1145 –1150,.
- [76] U. Kiencke and L. Nielsen: *Automotive Control Systems: For Engine, Driveline, and Vehicle*. (2000) . New York: Springer.
- [77] V. Alberti and E. Babbel: *Improved driving stability by active braking on the individual wheel*. (1996). In Proc. Int. Symp. Adv. Veh. Control, pp. 717–732.
- [78] Virginia Bernal Carretero: *Evaluación del comportamiento de sistemas reductores de*

velocidad para la regulación del tráfico. (2009) Universidad Carlos III de Madrid.

- [79] W. Klier, A. Reim, and D. Stapel: *Robust estimation of vehicle sideslip angle—An approach w/o vehicle and tire models.* (2008). In Proc. SAE World Congr., Detroit, MI, 2008, Paper 2008-01-0582.
- [80] Y. Fukada: *Slip- angle estimation for vehicle stability control. Vehicle System Dynamics.* (1999). 32, 375–388.
- [81] Z. X. Zhu, R. Torisu, J. I. Takeda, E. R. Mao and Q. Zhang: *Neural network for estimating vehicle behavior on sloping terrain.* (2005). Biosystems Engineering 91, 4, 403–411.

14 ANEXO I

Estadísticas

En este anexo se puede encontrar la tabla de datos sobre la estadística de accidentes de tráfico que se ha utilizado y de la cual se ha extraído los datos mostrados en el Capítulo 2 de este documento.



Causas Muerte Tráfico	Año Vía	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	Distr. % 2001	Distr. % 2011
Salida de la vía	Urbana	164	148	146	158	123	108	87	87	69	62	68	17%	15%
	Interurbana	1737	1691	1707	1466	1386	1191	1063	975	834	690	578	38%	36%
	Total	1901	1839	1853	1624	1509	1299	1150	1062	903	752	646	34%	31%
Colisión frontal	Urbana	61	58	52	51	45	40	29	29	28	15	14	6%	3%
	Interurbana	772	793	843	693	688	618	501	443	342	335	322	17%	20%
	Total	803	851	895	744	733	658	530	472	370	350	336	15%	16%
Colisión latera y frontolateral	Urbana	215	213	225	228	183	163	149	152	100	87	75	22%	16%
	Interurbana	958	905	907	826	745	560	578	464	371	341	254	21%	16%
	Total	1173	1118	1132	1054	928	723	727	616	471	428	329	21%	16%
Colisión trasera y múltiple	Urbana	50	51	40	37	42	38	63	38	44	30	28	5%	6%
	Interurbana	332	378	364	303	284	332	259	208	179	197	163	7%	10%
	Total	382	429	404	340	326	370	322	246	223	227	191	7%	9%
Atropello a peatón	Urbana	372	341	360	333	326	281	286	240	268	270	217	38%	47%
	Interurbana	445	420	407	319	334	284	262	210	191	179	150	10%	9%
	Total	817	761	767	652	660	565	548	450	459	449	367	15%	18%
Vuelco	Urbana	14	18	11	20	20	19	22	21	15	18	9	1%	2%
	Interurbana	83	74	71	70	46	57	64	41	55	48	38	2%	2%
	Total	97	92	82	90	66	76	86	62	70	66	47	2%	2%
Otro tipo de accidente	Urbana	98	83	85	73	51	88	105	67	60	68	46	10%	10%
	Interurbana	216	174	181	164	169	325	355	125	158	138	98	5%	6%
	Total	314	257	266	237	220	413	460	192	218	206	144	6%	7%
Total	Urbana	974	912	919	900	790	737	741	634	584	550	457	100%	100%
	Interurbana	4543	4435	4480	3841	3652	3367	3082	2466	2130	1928	1603	100%	100%
	Total	5517	5347	5399	4741	4442	4104	3823	3100	2714	2478	2060	100%	100%



15 ANEXO II

Software

En este anexo se presentan unos tutoriales sobre los Software CarSim y Java-NNS que se han utilizado para la realización de este proyecto.



15.1 CARSIM

A continuación se llevará a cabo una visión general del programa, haciendo mayor hincapié en aquellos menús de interés para este estudio. La versión de CarSim utilizada en este proyecto ha sido: 8.02.

En la Figura 15.1-1 se muestra la ventana principal del software con los menús principales resaltados. Estos se describirán a medida que se vaya avanzando en el texto y son:

- ❖ Menú 1: Permite definir el vehículo y sus características mecánicas.
- ❖ Menú 2: Permite configurar las maniobras de simulación y el entorno.
- ❖ Menú 3: Realizar la simulación.
- ❖ Menú 4: Visualizar el video o las gráficas de los parámetros pedidos.
- ❖ Menú 5: Permite superponer diferentes simulaciones.

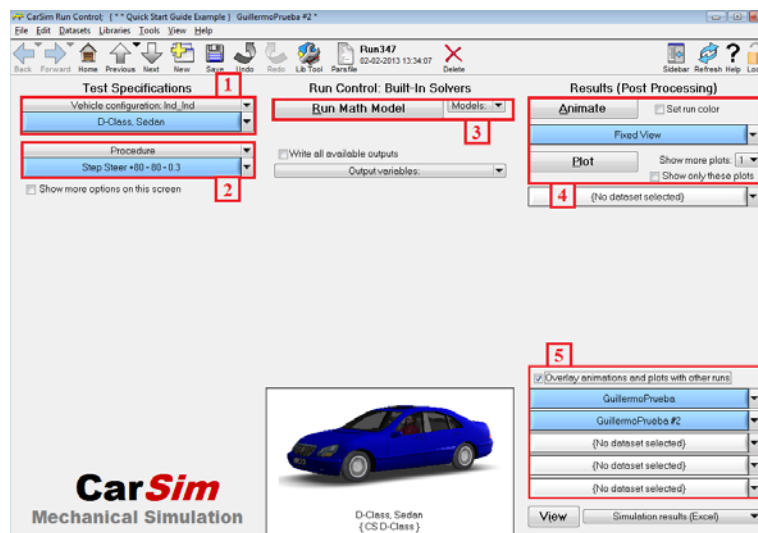


Figura 15.1-1: Ventana principal del software CarSim.

15.1.1 MENU 1: CONFIGURACIÓN DEL VEHÍCULO

CarSim da la posibilidad de seleccionar alguno de los vehículos incluidos en su librería y que han sido configurados de acuerdo a los parámetros convencionales de cada clase de vehículo, para ello, hay que pulsar el botón definido como “1.1”. No obstante, si ninguno de ellos se ajusta a lo deseado se puede acceder al submenú pulsando el botón “1.2” y configurar manualmente cada parámetro. Ver Figura 15.1.1-1.

En el submenú Figura15.1.1-2 se han resaltado las diferentes secciones. Cada una de esas secciones cuenta con valores predeterminados pero también se pueden modificar manualmente accediendo a cada submenú. Cada sección corresponde a:

- A). Dimensiones y masa del vehículo
- B). Parámetros relacionados con la aerodinámica
- C). Visualización del vehículo en las simulaciones
- D). Parámetros relacionados con la motorización
- E). Parámetros relacionados con el sistema de frenos
- F). Parámetros relacionados con el sistema de dirección
- G). Parámetros relacionados con el sistema de suspensión
- H). Parámetros relacionados con los neumáticos

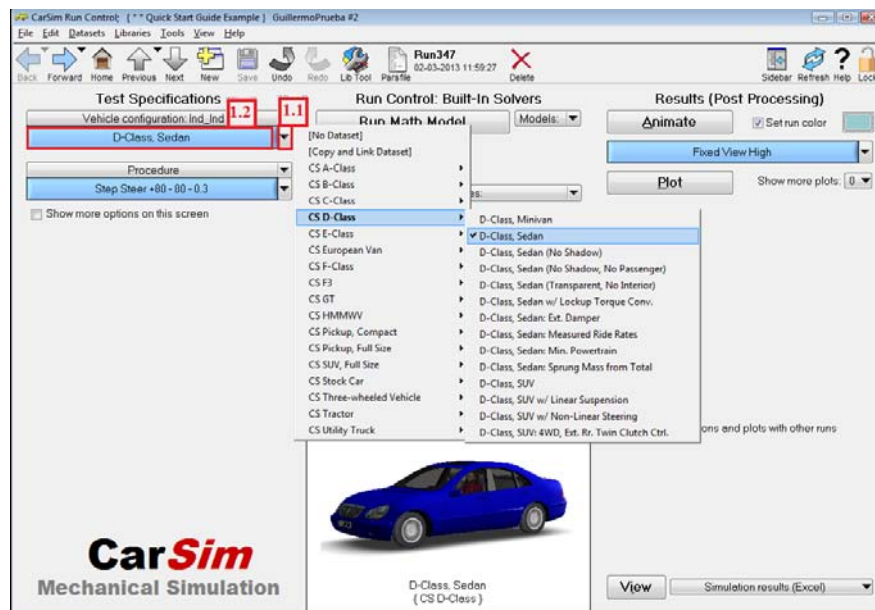


Figura 15.1.1-1: Menú configuración vehículo.

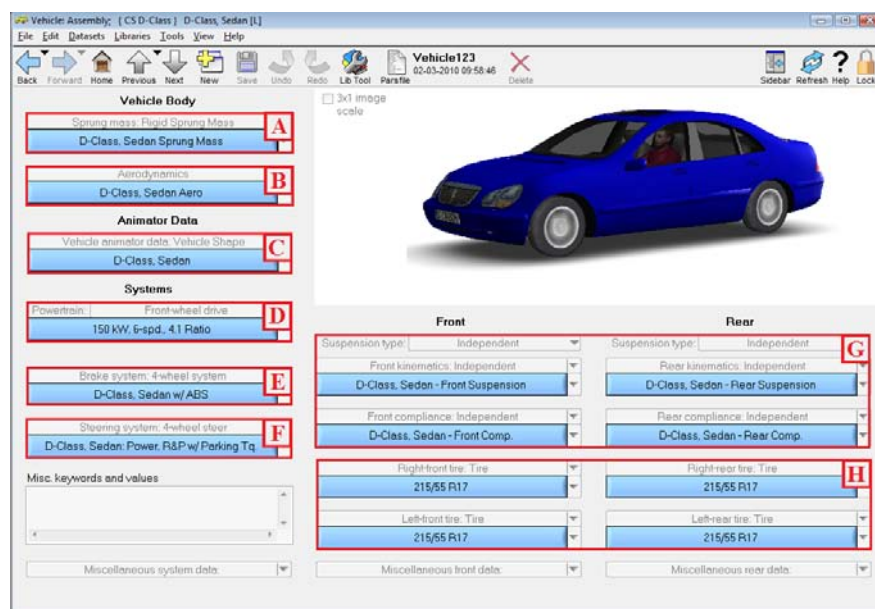


Figura 15.1.1-2: Submenú configuración vehículo.

15.1.2 MENÚ 2: CONFIGURACIÓN MANIOBRA Y ENTORNO

Análogamente al menú anterior, CarSim da la posibilidad de elegir alguna de las maniobras predeterminadas, pulsando el botón “2.1”, o acceder al submenú y configurar la maniobra que se quiera simular pulsando el botón “2.2”. Ver Figura 15.1.2-1.

En el submenú, Figura 15.1.2-2, se ha resaltado las distintas secciones de interés. Cada una de ellas cuenta con parámetros predeterminados pero se pueden modificar del mismo modo. Cada sección corresponde a:

- Configurar la velocidad del vehículo
- Configurar el control de los frenos
- Configurar el control de la dirección
- Configurar el entorno de la simulación (Circuito, paisaje...)
- Configurar el comienzo y el fin de la simulación
- Qué parámetros se desean extraer y graficar

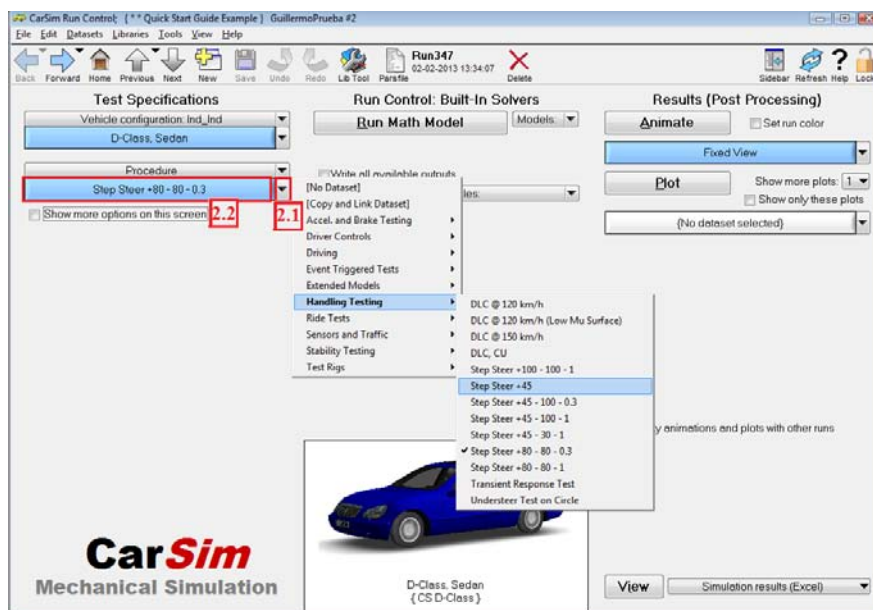


Figura 15.1.2-1: Menú configuración maniobra y entorno.

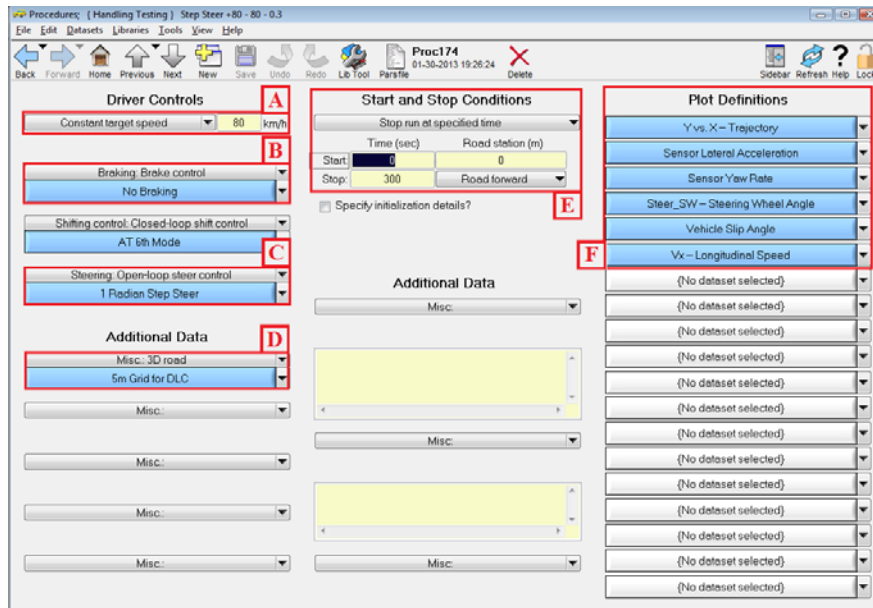


Figura 15.1.2-2: Submenú configuración maniobra y entorno.

Para la realización de las simulaciones de este proyecto se han modificado los parámetros incluidos en las secciones A, C y D:

- ❖ **Sección A:** Se configura la velocidad del vehículo durante la simulación. En el caso que ocupa, la velocidad será constante para cada maniobra.
- ❖ **Sección C:** Se configura la dirección del vehículo durante la maniobra. Ver Figura 15.1.2-3.
 - **C.1:** Se visualiza gráficamente los valores introducidos en C.2. La gráfica puede ser obtenida mediante alguno de los métodos dispuestos en el desplegable superior.
 - **C.2:** Se introducen en la tabla los valores de dirección para cada instante de tiempo. Posteriormente se da a calcular para actualizar la gráfica y cargar los valores. Los valores también pueden ser introducidos mediante un archivo Excel.
- ❖ **Sección D:** Se configura las características del entorno. Ver Figura 15.1.2-4
 - **D.1:** Se configura la geometría del circuito. Se ha elegido una superficie plana horizontal.
 - **D.2:** Se configura el coeficiente de rozamiento entre los neumáticos del vehículo y la calzada.
 - **D.3:** Se configura la visualización del entorno colocando árboles, conos, casas, paisaje...

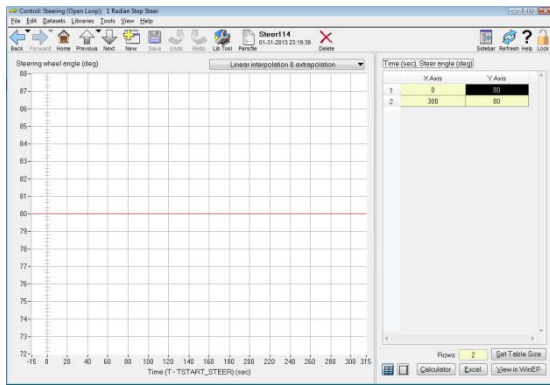


Figura 15.1.2-3: Configuración de la dirección.

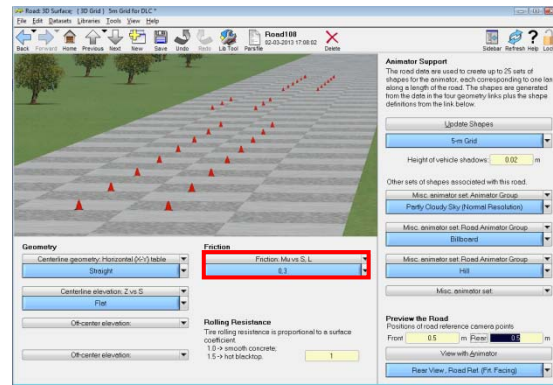


Figura 15.1.2-4: Configuración del coeficiente de adherencia.

15.1.3 MENÚ 3: REALIZAR LA SIMULACIÓN

Una vez que ya se ha configurado tanto el vehículo como la maniobra, se elige la opción con la que se quiere resolver el modelo y se pulsa el botón “Run Math model”. Ver Figura 15.1-1.

15.1.4 MENÚ 4: VISUALIZACIÓN DE LA SIMULACIÓN Y DE LOS PARÁMETROS

Cuando el modelo ha sido resuelto se puede visualizar la animación de la simulación pulsando el botón “Animate”, o bien, se pueden mostrar las gráficas de los parámetros pedidos en el menú 2 sección “F”, pulsando el botón “Plot”. Ver Figura 15.2-1 y 15.1.4-1.

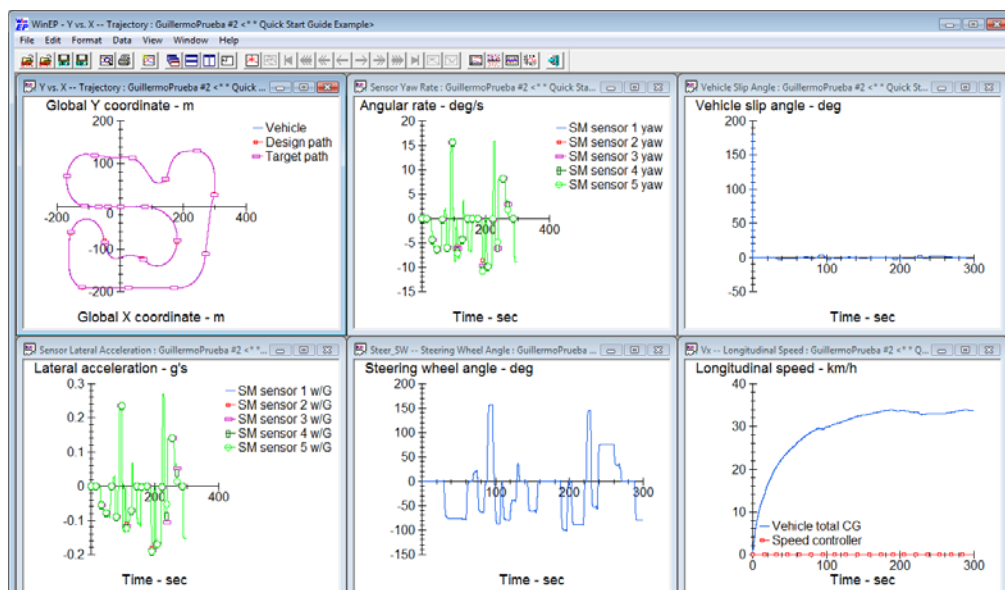


Figura 15.1.4-1: Visualización de las variables de la dinámica del vehículo tras la simulación.

15.1.5 MENÚ 5: SUPERPONER SIMULACIONES

CarSim cuenta con la posibilidad de realizar diferentes simulaciones y superponerlas tanto en el video como en las gráficas para ver como varía el comportamiento del vehículo o los parámetros extraídos. Esto se lleva a cabo en la sección 5 mostrada en la Figura 15.1-1.

15.2 JAVA-NNS

A continuación se ha llevado a cabo un tutorial básico sobre cómo manejar Java-NNS para realizar un entrenamiento y validación de una red neuronal. Se hará mayor hincapié en la configuración de la función de aprendizaje Backpropagation pues es el que se ha usado para llevar a cabo las simulaciones del presente proyecto.

Los pasos a seguir para realizar una simulación se enumeran a continuación y se desarrollarán posteriormente.

- ❖ Crear la tipología de la red.
- ❖ Crear las conexiones entre neuronas.
- ❖ Cargar patrones de entrenamiento y de validación.
- ❖ Configurar entrenamiento.
- ❖ Extraer los datos y gráficas.
- ❖ Lanzar entrenamiento.
- ❖ Guardar entrenamiento.
- ❖ Analizar datos con la red entrenada.

15.2.1 CREAR LA CONFIGURACIÓN DE LA RED

Tal y como se explicó en el apartado 7.3, una red neuronal está constituida por tres tipos de capas diferentes (entrada, oculta y salida). En este punto es cuando se definen cada una de las capas y la función de activación de cada una de las neuronas que la componen. Para ello se abre la siguiente ventana desde [Tools / Create / Layers]. Ver figura 15.2.1-1.

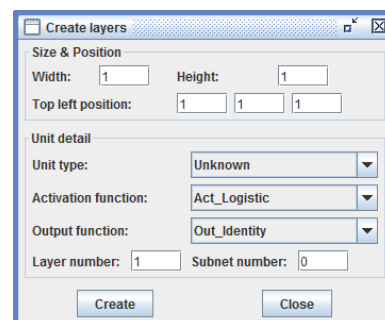


Figura 15.2.1-1: Create layers

A continuación se explican los principales campos que aparecen en la ventana:

- ❖ **Width:** Número de columnas de la capa.
- ❖ **Height:** Número de filas de la capa.
- ❖ **Unit type:** Tipo de capa.
 - Capa de entrada: Input.
 - Capas ocultas: Hidden.
 - Capa de salida: Output.
- ❖ **Activation function:** Función de activación. Ver Apartado 7.4.2.
 - Capa de entrada: función tangencial hiperbólica.
 - Capas ocultas: función tangencial hiperbólica.
 - Capa de salida: función identidad.
- ❖ **Layer Number:** Número de la capa que se está configurando.

En la Figura 15.2.1-2 se muestra una red ejemplo y los valores introducidos para cada capa.

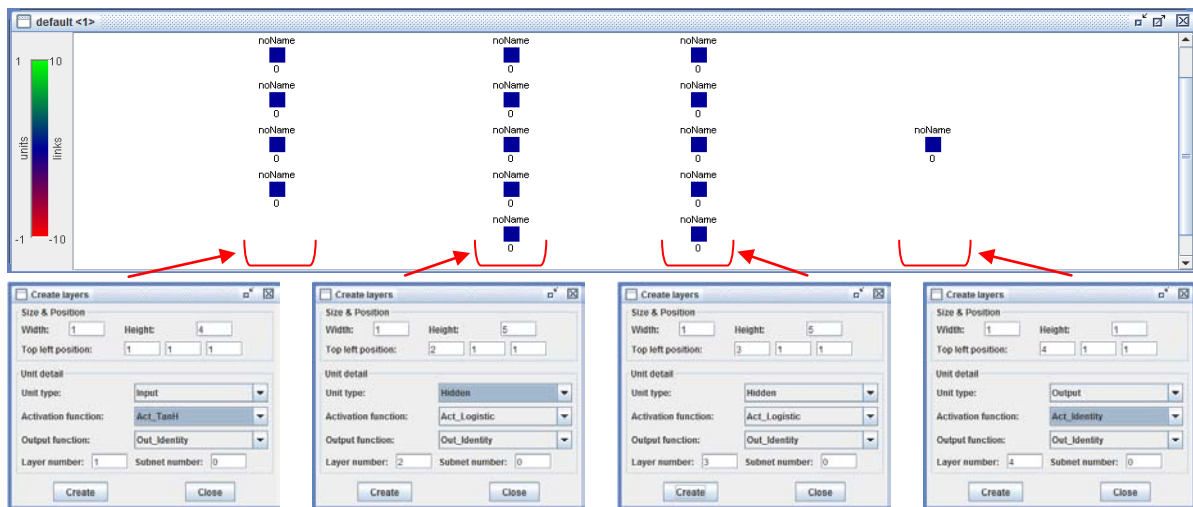


Figura 15.2.1-2: Creación de las neuronas de cada capa de una red.

Las neuronas pueden ser posteriormente editadas pulsando el botón derecho del ratón sobre ellas y seleccionando la opción [Edit] o ir a [Edit / Unit properties]. En la Figura 15.2.1-3 se muestra la ventana que se abre. En ella se puede cambiar los nombres de las neuronas y las funciones de activación.

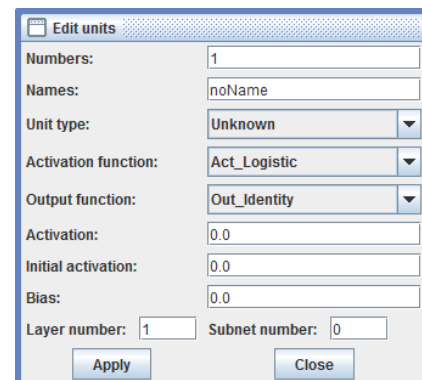


Figura 15.2.1-3: Ventana editar neuronas

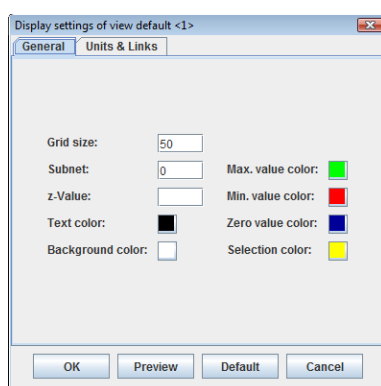


Figura 15.2.1-4: Ventana editar red

También se pueden editar las etiquetas que se verán para cada neurona y seleccionar si se desean ver links, pesos, etc. Para ello, ir a [View / Display setting] donde se abre la siguiente ventana. Ver Figura 15.2.1-4.

15.2.2 CREAR LAS CONEXIONES

Una vez que se tienen creadas las capas con las neuronas es necesario conectarlas. En este caso se han unido cada neurona con todas las neuronas de la siguiente capa. Desde [Tools / Create / Connections] se abre el menú mostrado en la Figura 15.2.2-1

En [File / Save as] se puede guardar la red con la extensión “.net”.

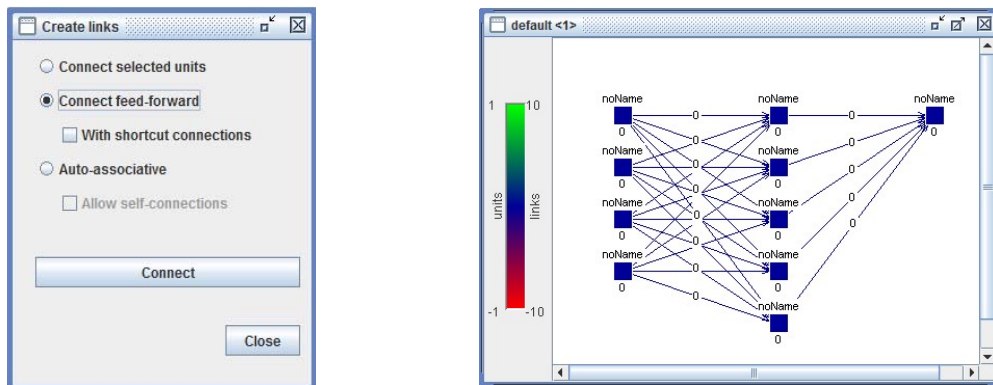


Figura 15.2.2-1: Crear conexiones de las neuronas.

15.2.3 CARGAR PATRONES

En el apartado 10.2 se han obtenido los datos con los que crear los patrones de entrenamiento y validación de la red. Estos patrones deben estar en un documento de texto donde cada fila corresponde a un patrón en el que se puede leer por columnas las variables de entrada y de salida, separadas por un solo espacio. Además, deberá tener el encabezado indicado en la Figura 15.2.3-1, en el que habrá que variar los valores de “No of patterns” (número de patrones), “input unit” (número de entradas) y “output units” (número de salidas) para que sean consistentes con el fichero que se quiere cargar.

```

SNN3 pattern definition file V3.2 generated at Mon Apr 25 18:03:29 1994

No. of patterns : 9624
No. of input units : 4
No. of output units : 1

0.225048981 0 0.3 -0.058546923 0
0.225042002 0.007208984 0.3 0.196374862 0.00743822
0.22491876 0.04186075 0.3 0.226635052 0.030476568
0.224775996 0.108768554 0.3 0.204365991 0.051473618
0.22458853 0.154132103 0.3 0.172260932 0.067630136
0.224361111 0.108600007 0.3 0.132362865 0.079159736
0.224102203 0.107731409 0.3 0.095313361 0.087199959
0.223857605 0.212882824 0.3 0.07456571 0.093333767
0.22351461 0.208526083 0.3 0.068669722 0.099360967
0.22320018 0.199872877 0.3 0.071732865 0.104217714
0.222881414 0.189773611 0.3 0.075640522 0.113776655
0.222545064 0.181650037 0.3 0.076898597 0.121169752
0.222286869 0.176332524 0.3 0.075408967 0.12741316
0.221993909 0.175342194 0.3 0.07251329 0.1316166
0.221677034 0.175542764 0.3 0.070732668 0.135316387
0.22140886 0.181787267 0.3 0.070785913 0.133566857
0.221154829 0.186664663 0.3 0.07180005 0.132446788
0.2209138 0.190273813 0.3 0.072496573 0.130836668
0.220684249 0.192618954 0.3 0.072027272 0.129259227
0.220464477 0.193158567 0.3 0.070457775 0.127795314
0.220282896 0.192832189 0.3 0.06845736 0.126707971
0.220048084 0.191152357 0.3 0.066799545 0.125961829
0.219848905 0.189556495 0.3 0.065980215 0.12551282
0.219654567 0.188081306 0.3 0.064052653 0.125401741
0.219464594 0.187013551 0.3 0.066729364 0.125476805
0.219270756 0.186397154 0.3 0.067595529 0.125685421
0.219097105 0.186178795 0.3 0.068315579 0.125945576
0.218919738 0.186241857 0.3 0.068726652 0.126190762
0.218746885 0.186454126 0.3 0.068833792 0.126381628

```

Figura 15.2.3-1: Documento de patrones.

Después de crear el fichero con el formato explicado, se abren en Java-NNS desde [File / open] y se guardan en [File / Save Data] usando la extensión “.pat”. Una vez generados los ficheros “.pat” con los patrones se vuelven a abrir desde [File / Open] para cargarlos finalmente en la red.

15.2.4 CONFIGURAR ENTRENAMIENTO

Para configurar la simulación es necesario abrir el panel de control ubicado en [Tools / Control panel], ver Figura 15.2.4-1. A continuación se detalla cada una de las pestañas que contiene.

- ❖ **Initializing:** Se inicializan los pesos de las conexiones creadas pulsando en el botón “init” tras haber introducido los parámetros $\min = -0.1$ y $\max = 0.1$.
- ❖ **Updating:** En esta pestaña se configura el orden para calcular las activaciones. Lo mejor es usar “Topological order” ya que con ello calcula primero las activaciones para la capa de entrada, luego para las ocultas, según su orden en la red, y por último, las de la capa de salida.
- ❖ **Patterns:** Se elige cual de los que documentos que contienen los patrones y que fueron cargados anteriormente será utilizado para el entrenamiento y la validación de la red neuronal. En la barra superior del programa aparecerán ambos patrones cargados.
- ❖ **Learning:** Pestaña dedicada a la elección del algoritmo de entrenamiento y los parámetros. En el caso que nos ocupa se ha elegido el algoritmo Backpropagation para realizar el entrenamiento de las diferentes configuraciones de redes.
 - **η :** define el tamaño de los saltos que dará el algoritmo para buscar el mínimo en la función del error. Habrá que irlo variando.
 - **dmax:** indica el valor del error que ya no es retropropagado. Es habitual seleccionar 0.1. El error retropropagado “dj”, en ese ciclo está definido a partir del error de la neurona de salida, “j” -calculado como, “ $dj=tj-oj$ ”- y del valor de dmax introducido, según lo siguiente:
 - Si $dj < dmax \rightarrow dj = 0$
 - Si $dj > 1-dmax \rightarrow dj = 1$
 - **Cycles:** número de épocas de entrenamiento. Después de realizarlas se hace una validación.
 - **Steps:** dejarlo en 1.
 - **Shuffle:** indica que los patrones pasarán de forma aleatoria cuando se hace entrenamiento uno a uno.
 - **Init:** pulsar para inicializar el entrenamiento.

- **Learn current:** realiza entrenamiento pasando los patrones uno a uno. Cada vez que pasa un patrón propaga errores hacia atrás.
- **Learn all:** realiza entrenamiento “batch”, es decir, pasan todos los patrones y luego propaga errores.

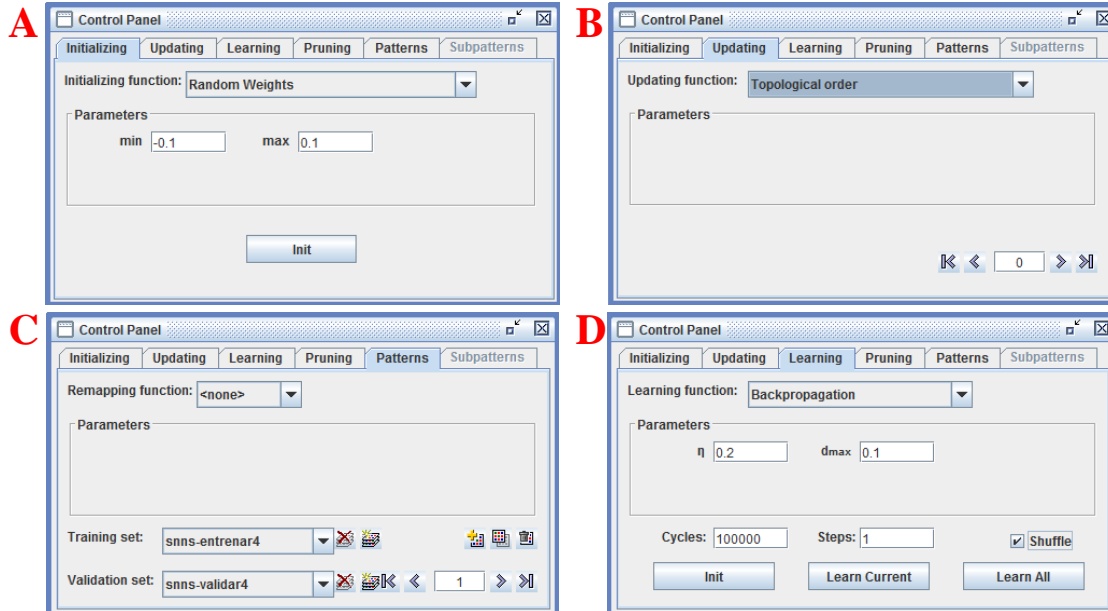


Figura 15.2.4-1: Configuración entrenamiento: A) Initializing. B) Updating. C) Patterns. D) Learning

15.2.5 EXTRAER LOS DATOS Y GRÁFICAS

Antes de iniciar el entrenamiento de la red es necesario abrir diferentes ventanas donde se registrarán todos aquellos datos y gráficas que posteriormente se quieran guardar. Ver Figura 15.2.5-1.

❖ **Gráfica de errores:** Se abre en [View / Error graph]. Dependiendo de cuál de los tres errores se seleccione en el margen izquierdo se estará visualizando uno de los siguientes: Ver apartado 7.8.

- $\sum e^2$: Se visualizan los errores SSE.
- $\frac{1}{n} \sum e^2$: Se visualizan los errores MSE.
- $\frac{1}{\text{outputs}} \sum e^2$: Se visualizan los errores dividido entre el número de neuronas de salida.

❖ **Gráfico de pesos:** Se abre en [View / Linkweights]. Se puede observar cómo varían los pesos.

- ❖ **Log:** Se registra el error obtenido en cada step, tanto para el entrenamiento como para la validación de los patrones. El tipo de error mostrado es el que se haya seleccionado en ese momento en la gráfica de errores.

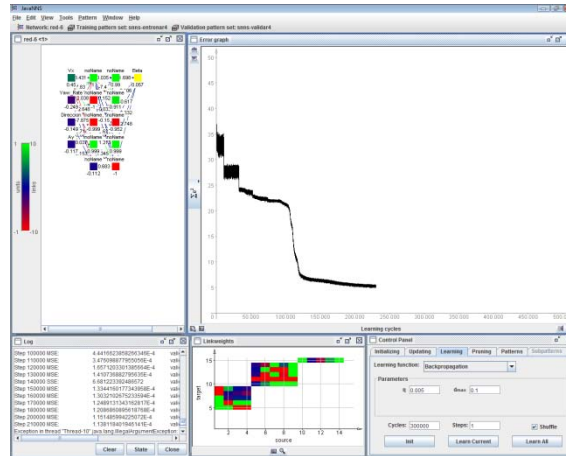


Figura 15.2.5-1: Ventana principal de simulación de una red.

15.2.6 LANZAR ENTRENAMIENTO

Para empezar a entrenar la red pulsar una de las dos opciones: Learn Current o Learn All explicadas con anterioridad en el apartado 15.2.4.

15.2.7 GUARDAR ENTRENAMIENTO

En [File / Save as] puede guardarse la red entrenada con la extensión “.net”. Si este documento se abre con un editor de texto se puede visualizar los pesos, las activaciones...

En [File / save data] pueden guardarse los siguientes archivos.

- ❖ **Archivo .res:** Permite guardar los valores estimados de las neuronas de salida, además permite adjuntar también los valores que han sido introducidos en los patrones. Para ello, aparece la siguiente ventana. Ver Figura 15.2.7-1.

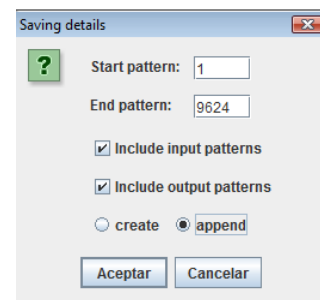


Figura 15.2.7-1: Guardar archivo .RES.

- ❖ **Archivo .pat:** Tiene la misma función que el anterior.
- ❖ **Archivo .log:** Se guardan los datos recogidos en la ventana LOG que se abrió.

15.2.8 ANALIZAR DATOS CON LA RED ENTRENADA

Una vez que la red ha sido entrenada se puede analizar con otro patrón de datos. Para ello, seguir los siguientes pasos:

- 1º. Abrir el documento con los patrones a analizar. Ver apartado 15.2.3
- 2º. Cargar el documento tanto para entrenamiento como para validación. Ver apartado 15.2.4.
- 3º. Escribir en el campo situado en la parte inferior derecha del panel de control el número de patrones que tiene el documento que se ha cargado. Ver Figura 15.2.8-1.

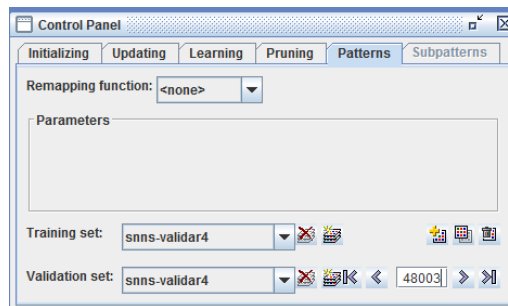


Figura 15.2.8-1: Configuración panel de control para análisis de la red entrenada.

- 4º. Configurar el análisis desde [Tools / Analyzer]. En la parte inferior existen tres pestañas que se han de configurar de la siguiente manera:

- ❖ Pestaña Setup. Ver Figura 15.2.8-2.
 - X-axis: Seleccionar en el desplegable “Pattern no”.
 - Y-axis:
 - Seleccionar en el desplegable “Unit”
 - Escribir en el campo el número de la neurona que se quiere analizar. En este caso la neurona de salida.
 - Seleccionar “Output” para analizar los datos de salida de la neurona.

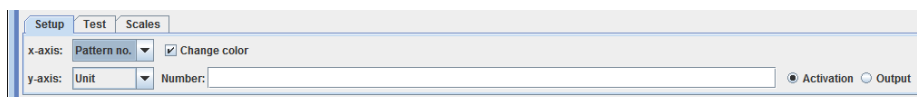


Figura 15.2.8-2: Pestaña setup para análisis de la red entrenada.

❖ Pestaña Test. Ver Figura 15.2.8-3

- En Steps escribir el número de patrones que se quieren analizar.
Si se quiere realizar el análisis completo se deberá escribir el mismo número que el introducido en el Panel de control.
- Mantener activo “Change pattern”.

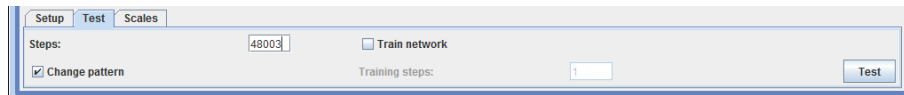


Figura 15.2.8-3: Pestaña test para análisis de la red entrenada.

5º. Realizar Análisis: Pulsar en el botón “Test”. Ver Figura 15.2.8-4



Figura 15.2.8-4: Análisis de una red entrenada.

6º. Guardar los datos: Los datos obtenidos en el análisis han sido registrados en la ventana LOG, por ello, desde [File / Save data] pueden ser guardados.